VERIFICATION OF TRANSLATION

I, Yuki Takahashi,

Of 4th Floor, MATRICE 2-13-7, MINAMIAOYAMA MINATO-KU, TOKYO, 107-0062, JAPAN,

am the translator of the documents attached and I state that the following is a true translation to the best of my knowledge and belief of Japanese Patent Application No. 2002-104581.

Dated this on June 4, 2009

Yuki Takahashi

Signature of translator

翻 訳 納 品 書

2009年6月4日

ソニー株式会社
知的財産センター　企画管理部
知的財産管理２Gp
前田　朋子　様

南青山国際特許事務所
翻訳担当　高橋　有紀
TEL: 03-5412-0315
FAX: 03-5412-0316

貴社ケース番号　　S03P0391US00
弊所ケース番号　　09NH046

翻訳内容　　：■　優先権証明書宣誓翻訳（JP2002-104581）

拝啓　貴社益々ご清栄のこととお慶び申し上げます。
同封にて、下記書類を納品させて頂きますのでご査収下さい。

敬具

納品物：
□　英文翻訳明細書（ワード／PDF）
□　外国図面（PDF）
□　優先権証明書（社内便のみ）
□　優先権証明書宣誓翻訳（オリジナル）
（控えとしてDOX経由で宣誓書（PDF）、優先権翻訳（ワード・PDF）納品済み）
□　その他

以上

[Name of the Document]	Request
[Reference No.]	0290376304
[Date of Filing]	April 5, 2002
[Address]	Commissioner of Patent Office
[IPC]	H03M 13/27

[Inventor]

[Address] c/o Sony Corporation
7-35, Kitashinagawa 6-chome, Shinagawa-ku, Tokyo

[Name] Mototsugu Takamura

[Applicant]

[ID No.] 000002185

[Name] c/o Sony Corporation

[Agent]

[ID No.] 100110434

[Patent Attorney]

[Name] Masaru Sato

[Indication of Charge]

[Ledger Account No.] 076186

[Amount of Payment] 21,000 yen

[List of Document Filed]

[Document]	Specification	1
[Document]	Drawings	1
[Document]	Abstract	1

[General Power of Attorney ID No.] 0011610

[Necessity of proof] Necessary

[NAME OF DOCUMENT] SPECIFICATION

[TITLE OF THE INVENTION] INTERLEAVING DEVICE, INTERLEAVING METHOD, DE-INTERLEAVING DEVICE, AND DE-INTERLEAVING METHOD

[SCOPE OF CLAIMS]

[CLAIM 1] An interleaving device replacing and rearranging an order of input data according to a predetermined address for output, characterized by comprising:

a first interleaving means for performing folding interleaving on first data constituted of plural input packets, in units of a data word or plural consecutive data words; and

a second interleaving means for performing folding interleaving on second data constituted of plural packets generated by said first interleaving means, in units of the packet.

[CLAIM 2] The interleaving device according to claim 1,

characterized in that said second interleaving means inverts a value of beginning data in a first packet of packet-unit folding interleave.

[CLAIM 3] The interleaving device according to claim 2,

characterized in that said beginning data is a sink byte in a header of a packet that constitutes said first data.

[CLAIM 4] The interleaving device according to claim 1, characterized in that:

said first interleaving means is installed by using first storage means incorporated in a programmable device; and

said second interleaving means is installed by using second storage means externally attached to said programmable device.

[CLAIM 5] The interleaving device according to claim 4,

characterized in that said first storage means is a dual-port random access memory in which inconsecutive addresses are accessed in units of a data word at each clock synchronized with the data word.

[CLAIM 6] The interleaving device according to claim 4,

characterized in that said second storage means is a random access memory fitted for burst transfer of data in units of plural data words.

[CLAIM 7] The interleaving device according to claim 6,

characterized in that said second storage means is a synchronous dynamic random access memory.

[CLAIM 8] The interleaving device according to claim 1,

characterized in that a predetermined error correction code is added to each of the packets that constitute said first data.

[CLAIM 9] The interleaving device according to claim 8,

characterized in that each of the packets that constitute said first data is obtained by adding said error correction code to plural transport packets, respectively, that constitute a transport stream obtained by performing compression and encoding on the basis of an MPEG-2 standard on predetermined data.

[CLAIM 10] The interleaving device according to claim 8,

characterized in that said error correction code is a Reed-Solomon code.

[CLAIM 11] The interleaving device according to claim 1,

characterized in that said first interleaving means performs the folding interleaving in units of a byte or plural consecutive bytes.

[CLAIM 12] An interleaving method replacing and rearranging an order of input data according to a predetermined address for output, characterized by comprising:

a first interleaving step of performing folding interleaving on first data constituted of plural input packets, in units of a data word or plural consecutive data words; and

a second interleaving step of performing folding interleaving, in units of a packet, on second data constituted of plural packets generated by the first interleaving step.

[CLAIM 13] The interleaving method according to claim 12,

characterized in that in said second interleaving step, a value of beginning data in a first packet of packet-unit folding interleave is inverted.

[CLAIM 14] The interleaving method according to claim 13,

characterized in that said beginning data is a sink byte in a header of a packet that constitutes said first data.

[CLAIM 15] The interleaving method according to claim 12, characterized in that:

said first interleaving step is performed by using first storage means which is incorporated in a programmable device; and

said second interleaving step is performed by using second storage means which is externally attached to the programmable device.

[CLAIM 16] The interleaving method according to claim 15,

characterized in that as said first storage means, a dual-port random access memory in which inconsecutive addresses are accessed in units of a data word at each clock synchronized with the data word is used.

[CLAIM 17] The interleaving method according to claim 15,

characterized in that as said second storage means, a random access memory fitted for burst transfer of data in units of plural data words is used.

[CLAIM 18] The interleaving method according to claim 17,

characterized in that as said second storage means, a synchronous dynamic random access memory is used.

[CLAIM 19] The interleaving method according to claim 12,

characterized in that a predetermined error correction code is added to each of the packets that constitute said first data.

[CLAIM 20] The interleaving method according to claim 19,

characterized in that each of the packets that constitute said first data is obtained by adding said error correction code to each of plural transport packets that constitute a transport stream obtained by performing compression and encoding on the basis of an MPEG-2 standard on predetermined data.

[CLAIM 21] The interleaving method according to claim 19,

characterized in that said error correction code is a Reed-Solomon code.

[CLAIM 22] The interleaving method according to claim 12,

characterized in that in said first interleaving step, the folding

interleaving is performed in units of a byte or plural consecutive bytes.

[CLAIM 23] A de-interleaving device replacing and rearranging an order of input data according to a predetermined address for output in such a manner as to restore a rearranged data array using an interleaving device including a first interleaving means for performing folding interleaving on first data constituted of plural input packets, in units of a data word or plural consecutive data words, and a second interleaving means for performing folding interleaving on second data constituted of plural packets generated by said first interleaving means, in units of the packet. the de-interleaving device being characterized by comprising:

a first de-interleaving means for performing folding de-interleaving, in units of a packet, on third data constituted of plural input packets, in such a manner as to restore an order of packets of data subjected to the folding interleaving by the second interleaving means to an order of the packets that constitute said second data; and

second de-interleaving means for performing folding de-interleaving, in units of a data word or plural consecutive data words, on fourth data constituted of plural packets generated by said first de-interleaving means, in such a manner as to restore an order of the packets that constitute said second data subjected to the folding interleaving by the first interleaving means to an order of the packets that constitute said first data.

[CLAIM 24] The de-interleaving device according to claim 23, characterized in that:

a value of beginning data in a first packet in packet-unit folding interleave is inverted by said second interleaving means; and

said first de-interleaving means synchronizes a first packet in said third data on the basis of the beginning data whose value has been inverted by said second interleaving means.

[CLAIM 25] The de-interleaving device according to claim 24,

characterized in that said beginning data is a sink byte in a header of a packet that constitutes said first data.

[CLAIM 26] The de-interleaving device according to claim 24,

characterized in that said first de-interleaving means restores the inverted value of said beginning data to an original value, to generate said fourth data.

[CLAIM 27] The de-interleaving device according to claim 23,

characterized in that said first de-interleaving means grasps whether a packet is lost on the basis of information indicating consecutiveness of packets that constitute said third data.

[CLAIM 28] The de-interleaving device according to claim 27,

characterized in that said first de-interleaving means replaces, when it is grasped that a packet is lost, data that corresponds to the lost packet with invalid data, to generate said fourth data.

[CLAIM 29] The de-interleaving device according to claim 23, characterized in that:

said first de-interleaving means is installed by using first storage means which is externally attached to a programmable device; and

said second de-interleaving means is installed by using second storage means which is incorporated in the programmable device.

[CLAIM 30] The de-interleaving device according to claim 29,

characterized in that said first storage means is a random access memory that is fitted for burst transfer of data in units of plural data words.

[CLAIM 31] The de-interleaving device according to claim 30,

characterized in that said first storage means is a synchronous dynamic random access memory.

[CLAIM 32] The de-interleaving device according to claim 29,

characterized in that said second storage means is a dual-port random access memory in which inconsecutive addresses are accessed in units of a data word at each clock synchronized with the data word.

[CLAIM 33] The de-interleaving device according to claim 23,

characterized in that a predetermined error correction code is added to each of the packets that constitute said first data.

[CLAIM 34] The de-interleaving device according to claim 33,

characterized in that each of the packets that constitute said first data is

obtained by adding said error correction code to plural transport packets, respectively, that constitute a transport stream obtained by performing compression and encoding on the basis of an MPEG-2 standard on predetermined data.

[CLAIM 35] The de-interleaving device according to claim 33, characterized in that said error correction code is a Reed-Solomon code.

[CLAIM 36] The de-interleaving device according to claim 23, characterized in that:

said first interleaving means performs the folding interleaving in units of a byte or plural consecutive bytes; and

said second de-interleaving means performs the folding de-interleaving in units of a byte or plural consecutive bytes.

[CLAIM 37] A de-interleaving method replacing and rearranging an order of input data according to a predetermined address for output in such a manner as to restore a rearranged data array using an interleaving method including a first interleaving step of performing folding interleaving on first data constituted of plural input packets, in units of a data word or plural consecutive data words, and a second interleaving step of performing folding interleaving, in units of a packet, on second data constituted of plural packets generated by the first interleaving step. the de-interleaving method being characterized by comprising:

a first de-interleaving step of performing folding de-interleaving, in units of a packet, on third data constituted of plural input packets, in such a manner as to restore an order of packets of data subjected to the folding interleaving in the second interleaving step to an order of the packets that constitute said second data; and

a second de-interleaving step of performing folding de-interleaving, in units of a data word or plural consecutive data words, on fourth data constituted of plural packets generated in said first de-interleaving step, in such a manner as to restore an order of the packets that constitute said second data subjected to the folding interleaving in the first interleaving step to an order of the packets that

constitute said first data.

[CLAIM 38] The de-interleaving method according to claim 37, characterized in that:

a value of beginning data in a first packet in packet-unit folding interleave is inverted in said second interleaving step; and

in said first de-interleaving step, a first packet in said third data is synchronized on the basis of the beginning data whose value has been inverted in said second interleaving step.

[CLAIM 39] The de-interleaving method according to claim 38,

characterized in that said beginning data is a sink byte in a header of a packet that constitutes said first data.

[CLAIM 40] The de-interleaving method according to claim 38,

characterized in that in said first de-interleaving step, the inverted value of said beginning data is restored to an original value, to generate said fourth data.

[CLAIM 41] The de-interleaving method according to claim 37,

characterized in that in said first de-interleaving step, whether a packet is lost is grasped on the basis of information indicating consecutiveness of packets that constitute said third data.

[CLAIM 42] The de-interleaving method according to claim 41,

characterized in that in said first de-interleaving step, when it is grasped that a packet is lost, data corresponding to the lost packet is replaced with invalid data, to generate said fourth data.

[CLAIM 43] The de-interleaving method according to claim 37, characterized in that:

said first de-interleaving step is performed by using first storage means which is externally attached to a programmable device; and

said second de-interleaving step is performed by using second storage means which is incorporated in said programmable device.

[CLAIM 44] The de-interleaving method according to claim 43,

characterized in that as said first storage means, a random access memory fitted for burst transfer of data in units of plural data words is used.

[CLAIM 45] The de-interleaving method according to claim 44,

characterized in that as said first storage means, a synchronous dynamic random access memory is used.

[CLAIM 46] The de-interleaving method according to claim 43,

characterized in that as said second storage means, a dual-port random access memory in which inconsecutive addresses are accessed in units of a data word at each clock synchronized with the data word is used.

[CLAIM 47] The de-interleaving method according to claim 37,

characterized in that a predetermined error correction code is added to each of the packets that constitute said first data.

[CLAIM 48] The de-interleaving method according to claim 47,

characterized in that each of the packets that constitute said first data is obtained by adding said error correction code to plural transport packets, respectively, that constitute a transport stream obtained by performing compression and encoding on the basis of an MPEG-2 standard on predetermined data.

[CLAIM 49] The de-interleaving method according to claim 47,

characterized in that the error correction code is a Reed-Solomon code.

[CLAIM 50] The de-interleaving method according to claim 37, characterized in that:

in said first interleaving step, the folding interleaving is performed in units of a byte or plural consecutive bytes; and

in said second de-interleaving step, the folding de-interleaving is performed in units of a byte or plural consecutive bytes.

[DETAILED DESCRIPTION OF THE INVENTION]

[0001]

[TECHNICAL FIELD TO WHICH THE INVENTION BELONGS]

The present invention relates to an interleaving device and an interleaving method for replacing and rearranging an order of input data according to a predetermined address for output, and a de-interleaving device and a de-interleaving method for replacing and rearranging an order of input data

according to a predetermined address for output, in such a manner as to restore a rearranged data array using the interleaving device and the interleaving method.

[0002]

[PRIOR ART]

In recent years, technologies have been developed for handling a video signal and/or an audio signal of a so-called HDTV (High Definition Television) such as a broadcast system formed by use of, for example, a broadcasting or business use video camera. In such a system, an apparatus called a transcoder, which combines a function to compress and encode non-compressed video and/or audio signals by using an image compression system such as so-called MPEG (Moving Picture Experts Group)-2 and a function to decompress and decode these compressed and encoded video and/or audio signals, is used. That is, this transcoder performs compression and encoding on the basis of such an image compression system as MPEG-2 by use of a mounted encoder on non-compressed video and/or audio signals input through a serial bus etc. which conforms to, for example, an HD-SDI (High Definition-Serial Digital Interface), and transmits them as a transport stream constituted of plural transport packets (TS packets). Further, the transcoder can decompress and decode, by using a mounted decoder, a transport stream received through a predetermined communication path, to reproduce non-compressed video and/or audio signals.

[0003]

As in the case of a transport stream transmitted and received by such a transcoder, in packet communication in which predetermined data is transmitted and received with it being stored in a packet, a header is added to each of the packets, so that an overhead can be reduced as larger size data is transferred with it being stored in one packet, thus being effective in terms of a communication efficiency.

[0004]

Further, in packet communication, data is transferred in accordance with the IP (Internet Protocol) often, so that packets are transmitted and received in accordance with a variety of transfer protocols that match a data format etc.

Generally, packet communication based on the IP, as in the case of FTP (File Transfer Protocol), for example, often conforms to a transfer protocol capable of re-transmitting a packet in accordance with a condition of the communication path. However, in recent years, there are cases where the packet communication is applied to an application for real-time communication in which. re-transmission of packets is not permitted, for example, an RTP (Real-time Transport Protocol) known as a transfer protocol for reproducing video and/or audio signals by streaming. In this case, a transmission side, which transmits a packet, needs to have a powerful error correction capability so that a reception side can correct a significant burst error involving a packet loss in which a packet is lost due to an effect such as noise occurring on the communication path.

[0005]

As such error-correcting technologies, an addition of a predetermined error correction code such as a so-called Reed-Solomon code to a packet is conceivable.

[0006]

Here, a description will be given on an example that uses a Reed-Solomon code as the error correction code. By a Reed-Solomon code, if a 28-byte error correction code is added to a packet constituted of, for example, 188 bytes, error can be corrected completely when the number of bytes containing an error in each packet is 14 or less. That is, a Reed-Solomon code is typically an error correction code that can only cope with a random error. However, a Reed-Solomon code is not capable of correcting errors if the number of bytes containing an error in each packet is in excess of 14 bytes. Therefore, a Reed-Solomon code has no effect on a significant burst error that contains a packet loss.

[0007]

In this regard, as the error-correcting technologies, generally, a predetermined error correction code such as a Reed-Solomon code is added to a packet and then folding interleaving is performed on it, to disperse errors.

[0008]

That is, in the packet communication, even if a burst error, which has occurred on a communication path, is superimposed on packets on which a transmission side has performed the folding interleaving, a reception side can perform the corresponding folding de-interleaving on them to handle each of them as a random error for each packet and decode an error correction code such as a Reed-Solomon code, thereby correcting even the burst error. For example, in the packet communication, if folding interleaving with a depth of 18 is performed on packets each constituted of 216 bytes which is obtained by adding a 28-byte error correction code to the above-mentioned 188-byte packet, even when one of the consecutive 18 packets has become an error completely, the reception side can correct the error completely. Therefore, in the packet communication, although consecutive pieces of information of one packet may be all lost especially when data is transmitted or received through a communication path on which a packet loss is apt to occur; by using such an error-correcting technology, the lost packet can be restored completely.

[0009]

In such a manner, in the packet communication, in a case where it is applied to an application for real-time communication in which re-transmission of packets is not permitted, the error correction capability for a burst error containing a packet loss can be improved by adding a predetermined error correction code to a packet and performing folding interleaving on a packet thus obtained.

[0010]

[PROBLEMS TO BE SOLVED BY THE INVENTION]

Incidentally, in the packet communication, to correct a significant burst error containing a packet loss by using an error-correcting technology that combines the above-mentioned error correction code and folding interleaving, it is necessary to use an error correction code having a very large code length that matches a packet size.

[0011]

However, in the packet communication, if an error correction code

having a large code length is added, a data overhead increases, thus inducing deterioration of communication efficiency. Further, in the packet communication, addition of such an error correction code with a large code length brings about not only deterioration in communication efficiency but also an increase in scale of circuits of an encoder and a decoder for an error correction code.

[0012]

Therefore, in the packet communication, it is desirable to reduce the code length of an error correction code as much as possible.

[0013]

On the other hand, in the packet communication, if the code length of the error correction code is reduced, a depth of folding interleaving must be increased largely to correct a significant burst error containing a packet loss.

[0014]

In terms of implementation of folding interleaving, a depth of the folding interleaving and a required memory capacity are proportional to each other. Therefore, in the packet communication, to make the depth of folding interleaving extremely large, the required memory capacity increases, which is not desirable.

[0015]

Further, in the folding interleaving, as the memory, it is necessary to use something capable of accessing inconsecutive addresses for each clock of about one clock synchronized with each data word in units of a data word like a byte in the case of, for example, the MPEG standard, that is, a random access dual-port RAM (Dual Port Random Access Memory; hereinafter, referred to as DPRAM).

[0016]

Here, as the folding interleaving, a dedicated integrated circuit can be mounted; but, in terms of costs, handling ease, etc., it is effective to mount a general-purpose programmable device such as a so-called PLD (Programmable Logic Device) or FPGA (Field Programmable Gate Array). However, in the folding interleaving, since a DPRAM provided in such a general-purpose

programmable device has a small capacity, it is difficult to obtain a sufficient depth necessary to correct a significant burst error containing a packet loss by compensating for a decrease in code length of an error correction code. Further, in the folding interleaving, if a problem is a memory capacity alone, the problem can be solved by using a general-purpose memory to be externally mounted to a programmable device such as an SDRAM (Synchronous Dynamic Random Access Memory); however, such an externally mounted device is surely effective in burst transfer of data in units of plural data words but needs a period for inputting a command signal for access, so temporally-consecutive random access of addresses in units of a data word is difficult because it increases an overhead and decreases a throughput. Moreover, to realize such access, since very complicated circuits must be provided in a periphery of the memory, it is difficult to mount them for the purpose of high-speed access in units of a data word.

[0017]

As described above, in the packet communication, to correct a significant burst error containing a packet loss by using an error-correcting technology that combines an error correction code and folding interleaving, there has been a problem that a deterioration in communication efficiency due to a use of an error correction code having a large code length and an increase in circuit scale are induced or an increase in memory capacity due to realization of deep folding interleaving instead of using an error correction code having a small code length is induced.

[0018]

The present invention has been made in view of the circumstances as described above, and it is an object of the present invention to provide an interleaving device and an interleaving method that can realize folding interleaving, which enables correction of a significant burst error containing a packet loss by using an error correction code having a small code length on a transmission side in packet communication, and a de-interleaving device and a de-interleaving method that can restore on a reception side an array of data on which the folding interleaving has been performed using the interleaving device

and the interleaving method.

[0019]

[MEANS FOR SOLVING THE PROBLEMS]

To attain the above object, according to the present invention, there is provided an interleaving device replacing and rearranging an order of input data according to a predetermined address for output, characterized by including: a first interleaving means for performing folding interleaving on first data constituted of plural input packets, in units of a data word or plural consecutive data words; and a second interleaving means for performing folding interleaving on second data constituted of plural packets generated by said first interleaving means, in units of the packet.

[0020]

In such an interleaving device according to the present invention, the folding interleaving in units of a data word or plural consecutive data words performed by the first interleaving means and the folding interleaving in units of a packet performed by the second interleaving means are performed successively.

[0021]

Further, to achieve the object above, according to the present invention, there is provided an interleaving method replacing and rearranging an order of input data according to a predetermined address for output, characterized by including: a first interleaving step of performing folding interleaving on first data constituted of plural input packets, in units of a data word or plural consecutive data words; and a second interleaving step of performing folding interleaving, in units of a packet, on second data constituted of plural packets generated by the first interleaving step.

[0022]

In such an interleaving method according to the present invention, the folding interleaving in units of a data word or plural consecutive data words and the folding interleaving in units of a packet are performed successively.

[0023]

Further, to achieve the object above, according to the present invention,

there is provided a de-interleaving device replacing and rearranging an order of input data according to a predetermined address for output in such a manner as to restore a rearranged data array using an interleaving device including a first interleaving means for performing folding interleaving on first data constituted of plural input packets, in units of a data word or plural consecutive data words, and a second interleaving means for performing folding interleaving on second data constituted of plural packets generated by said first interleaving means, in units of the packet, the de-interleaving device being characterized by including: a first de-interleaving means for performing folding de-interleaving, in units of a packet, on third data constituted of plural input packets, in such a manner as to restore an order of packets of data subjected to the folding interleaving by the second interleaving means to an order of the packets that constitute said second data; and second de-interleaving means for performing folding de-interleaving, in units of a data word or plural consecutive data words, on fourth data constituted of plural packets generated by said first de-interleaving means, in such a manner as to restore an order of the packets that constitute said second data subjected to the folding interleaving by the first interleaving means to an order of the packets that constitute said first data.

[0024]

In such a de-interleaving device according to the present invention, the folding de-interleaving in units of a packet performed by the first de-interleaving means and the folding de-interleaving in units of a data word or plural consecutive data words performed by the second de-interleaving means are performed successively with respect to data on which the folding interleaving in units of a data word or plural consecutive data words and the folding interleaving in units of a packet have been performed successively.

[0025]

Furthermore, to achieve the object above, according to the present invention, there is provided a de-interleaving method replacing and rearranging an order of input data according to a predetermined address for output in such a manner as to restore a rearranged data array using an interleaving method

including a first interleaving step of performing folding interleaving on first data constituted of plural input packets, in units of a data word or plural consecutive data words, and a second interleaving step of performing folding interleaving, in units of a packet, on second data constituted of plural packets generated by the first interleaving step, the de-interleaving method being characterized by including: a first de-interleaving step of performing folding de-interleaving, in units of a packet, on third data constituted of plural input packets, in such a manner as to restore an order of packets of data subjected to the folding interleaving in the second interleaving step to an order of the packets that constitute said second data; and a second de-interleaving step of performing folding de-interleaving, in units of a data word or plural consecutive data words, on fourth data constituted of plural packets generated in said first de-interleaving step, in such a manner as to restore an order of the packets that constitute said second data subjected to the folding interleaving in the first interleaving step to an order of the packets that constitute said first data.

[0026]

In such a de-interleaving method according to the present invention, the folding de-interleaving in units of a packet and the folding de-interleaving in units of a data word or plural consecutive data words are performed successively with respect to data on which the folding interleaving in units of a data word or plural consecutive data words and the folding interleaving in units of a packet have been performed successively.

[0027]

[EMBODIMENT MODE OF THE INVENTION]

Hereinafter, a specific embodiment to which the present invention is applied will be described in detail with reference to the drawings.

[0028]

In this embodiment, a data transmission/reception system includes an encoder for receiving packeted data, adding a predetermined error correction code to each of the packets, performing interleaving on them, and transmitting them, and a decoder for decoding incoming data received from this encoder through a

predetermined communication path. In this data transmission/reception system, the encoder performs, as the folding interleaving, folding interleaving in units of a data word or plural consecutive data words and interleaving in units of a packet, to enable a significant burst error containing a packet loss to be corrected using an error correction code having a small code length. In the data transmission/reception system, the decoder can, on the other hand, decode incoming data transmitted as encoded by such an encoder and received via a predetermined communication path and completely correct a significant burst error containing a packet loss.

[0029]

It should be noted that the following description is made on the assumption that in the data transmission/reception system, the encoder is input with plural transport packets (TS packets) that constitute a transport stream obtained by performing compression and encoding on the basis of the so-called MEPG-2 standard on non-compressed video and/or audio signals input through a serial bus etc. which conforms to, for example, an HD-SDI (High Definition-Serial Digital Interface) standards, and a so-called Reed-Solomon code is used as an error correction code to be added by this encoder to the transport packet. Moreover, for convenience, descriptions will be given assuming that, in the data transmission/reception system, the encoder performs folding interleaving in units of a data word and folding interleaving in units of a packet and the decoder performs folding de-interleaving in units of a packet and folding interleaving in units of a data word.

[0030]

As shown in FIG. 1, for example, a data transmission/reception system includes an encoder 10 for receiving a transport stream TS constituted of plural transport packets TSP and encoding it, and a decoder 20 for decoding receive data RD received from this encoder 10 through a predetermined communication path and restoring the transport stream TS.

[0031]

First, the encoder 10 will be described.

[0032]

As shown in the figure, for example, the encoder 10 has: a Reed-Solomon-encoding portion 11 for performing Reed-Solomon encoding on each of the transport packets TSP which constitute the transport stream TS; a byte interleaver 12 for performing folding interleaving, in units of a byte, on encoded data ED constituted of plural encoded packets EP to each of which an error correction code has been added by the Reed-Solomon encoder 11; a packet interleaver 13 for performing folding interleaving, in units of a packet, on byte interleave data BID constituted of plural byte interleave packets BIP on which the byte interleaver 12 has performed folding interleaving; a packeting portion 14 for generating one item of packet data PD by linking to each other predetermined numbers of packet-unit interleave packets PIP of packet interleave data PID constituted of the plural packet-unit interleave packets PIP on which the packet interleaver 13 has performed the folding interleaving in units of a packet; and a transmission portion 15 for adding a predetermined header to each packet data PD generated by the packeting portion 14 and transmitting it as transmit data TD constituted of plural upper-layer packets ULP.

[0033]

In the encoder 10, of these portions, at least the Reed-Solomon-encoding portion 11, the byte interleaver 12, an interface portion 17 in the packet interleaver 13, and the packeting portion 14, which are enclosed by a broken line in the figure , are mounted as a programmable device such as a so-called PLD (Programmable Logic Device) or FPGA (Field Programmable Gate Array).

[0034]

The Reed-Solomon-encoding portion 11 performs Reed-Solomon encoding on each of the transport packets TSP which constitute a transport stream TS input as an information series, to generate encoded data ED constituted of plural encoded packets EP. Specifically, the Reed-Solomon-encoding portion 11 adds an error correction code (ECC) constituted of 28 bytes as a parity, as shown in FIG. 2(B), to each of the transport packets TSP of a transport stream TS, each of which is constituted of 188 bytes as shown in FIG. 2(A), to generate encoded

data ED constituted of encoded packets EP each of which has a code length of 216 bytes per packet. Note here that beginning data of each of the encoded packets EP is a sink byte in a packet header of the transport packet TSP according to the MPEG-2 standard and has a value of "0x47". The Reed-Solomon-encoding portion 11 supplies the subsequent byte interleaver 12 with encoded data ED constituted of the generated plural encoded packets EP. Note here that an error correction capability in a case where this encoded data ED is output as it is enough to completely correct an error of up to 14 bytes for each encoded packet EP.

[0035]

The byte interleaver 12 performs folding interleaving having predetermined depth and period on each of the encoded packets EP of the encoded data ED supplied from the Reed-Solomon-encoding portion 11, in units of a data word, that is, in units of a byte due to the MPEG standard, replaces and rearranges an order of the bytes of each of the encoded packets EP, and thus generates a byte interleave data BID constituted of plural byte interleave packets BIP. Specifically, although not shown, the byte interleaver 12 is constituted of a dual-port RAM (Dual Port Random Access Memory; hereinafter, referred to as DPRAM) capable of accessing inconsecutive addresses for each clock synchronized with a byte for each of the bytes, an address generation portion for performing data write and read with respect to this DPRAM, etc. The byte interleaver 12 sequentially transfers items of input encoded data ED to the DPRAM in accordance with a predetermined write address and writes them to it, and sequentially reads the data thus written to this DPRAM in accordance with a predetermined read address that is different from the write address, thereby performing, for example, folding interleaving with a depth of 18 and a period of 12 in units of a byte on the encoded data ED generated by the Reed-Solomon-encoding portion 11.

[0036]

Here, a depth of folding interleaving indicates such a numerical value that packets as many as a first number of bytes are divided into groups as many as

a second number of bytes which is smaller than the first number of bytes, to disperse each of the bytes of a divided unit constituted of this second number of bytes into mutually different divided units, and a period indicates a separation quantity for the divided unit, by which the adjacent bytes are dispersed, and is represented as a quotient obtained by dividing the first number of bytes by a depth of folding interleaving.

[0037]

That is, as shown on the first row of FIG. 3, the byte interleaver 12 performs folding interleaving such that the first byte of the first divided unit constituted of 18 bytes in one encoded packet EP constituted of 216 bytes is not rearranged, the following second byte is rearranged to be positioned at a position delayed only by 18x12x1-1 bytes, that is, at the second byte of the second packet, and the following third byte is rearranged to be positioned at a position delayed only by 18x12x2-1 bytes, that is, at the third byte of the third packet, to thus generate byte interleave data BID constituted of plural byte interleave packets BIP. Therefore, because of folding interleaving being performed by the byte interleaver 12, information contained in one encoded packet EP is dispersed 12 bytes at a time into 18 consecutive byte interleave packets BIP.

[0038]

Accordingly, although this does not cause the byte interleaver 12 to improve the error correction capability of the Reed-Solomon-encoding portion 11, the information can be dispersed in arrangement to transform a burst error containing a packet loss into a random error. Specifically, by an error correction capability in a case where byte interleave data BID is output as it is, even if one packet per 18 consecutive byte interleave packets BIP, that is, 216 bytes are completely erroneous, all the bits can be corrected. Note here that beginning data of each of the byte interleave packets BIP is no different from that of the encoded packet EP and is a sink byte in the packet header of a transport packet TSP according to the MPEG-2 standard, having a value of “0x47”. The byte interleaver 12 supplies the subsequent packet interleaver 13 with byte interleave data BID constituted of plural byte interleave packets BIP thus generated.

[0039]

Note here that this byte interleaver 12 can control addresses in accordance with a predetermined address generation system to reduce a capacity of a memory, that is, the DPRAM, necessary to perform folding interleaving in units of a byte, the descriptions of which will be given later.

[0040]

The packet interleaver 13 performs folding interleaving with predetermined depth and period in units of a packet on each of the byte interleave packets BIP that constitute byte interleave data BID supplied from the byte interleaver 12, and rearranges the byte interleave packets BIP in terms of order, thereby generating packet interleave data PID constituted of plural packet-unit interleave packets PIP. Specifically, the packet interleaver 13 includes an SDRAM (Synchronous Dynamic Random Access Memory) 16 externally attached to each of the portions mounted as a programmable device in order to burst-transfer data in units of plural bytes, and the interface portion 17 for performing processing such as address generation and data transmission/reception in order to perform data write and read with respect to this SDRAM 16. The packet interleaver 13 sequentially transfers and writes items of the byte interleave data BID input to the interface portion 17, to the SDRAM 16 in accordance with a predetermined write address and sequentially reads the items of data written to this SDRAM 16 in accordance with a predetermined read address that is different from the write address, thereby performing folding interleaving with a depth of 6 and a period of 4 in units of byte interleave packets on byte interleave data BID supplied from the byte interleaver 12.

[0041]

That is, as shown on the first row of FIG. 4, the packet interleaver 13 regards 24 consecutive byte interleave packets BIP as one packet (hereinafter, referred to as temporary composite packet TCP) and performs such folding interleaving as to avoid rearranging the first byte interleave packet BIP that constitutes a divided unit constituted of the first six byte interleave packets BIP in one temporary composite packet TCP, rearrange the following second byte

interleave packet BIP at a position delayed only by 6x4x1-1 packets, that is, at the second packet in the second temporary composite packet TCP, and rearrange the following third byte interleave packet BIP at a position delayed only by 6x4x2-1 packets, that is, at the third packet in the third temporary composite packet TCP, to thus generate packet interleave data PID constituted of plural packet-unit interleave packets PIP. Therefore, because of folding interleaving being performed by the byte packet interleaver 13, information contained in one encoded packet EP is sporadically dispersed into 6x4(=24)x6=144 consecutive packet-unit interleave packets PIP in such a condition that the information 12 bytes each is dispersed with a spacing of at least six packet-unit interleave packets PIP therebetween. Note that the SDRAM 16 that constitutes the packet interleaver 13 only needs to have a capacity to store at least 144 packet-unit interleave packets PIP.

[0042]

Accordingly, although, as in the case of the byte interleaver 12, this does not cause the packet interleaver 13 to improve the error correction capability of the Reed-Solomon-encoding portion 11, by performing dispersion arrangement of the information more sporadically than the byte interleaver 12, a larger burst error containing a packet loss can be made a random error. Specifically, by an error correction capability in a case where packet interleave data PID is output as it is, even if up to six consecutive packet-unit interleave packets PIP out of 144 consecutive packet-unit interleave packets PIP, that is, 1296 bytes are completely erroneous, all the bits can be corrected.

[0043]

Here, the packet interleave data PID on which the folding interleaving has been performed by the packet interleaver 13 cannot be grasped on a reception side as it is, and synchronization thus becomes difficult. Therefore, to notify the reception side of the beginning packet in the packet interleave data PID, the packet interleaver 13 inverts a value of a sink byte which is beginning data of a first packet-unit interleave packet PIP. That is, since folding interleaving with a depth of 6 is performed, the packet interleaver 13 inverts the beginning data of a

packet-unit interleave packet PIP every six packets from "0x47" to "0xB8", thereby generating packet interleave data PID to be output eventually. The packet interleaver 13 supplies to the subsequent packeting portion 14 the packet interleave data PID constituted of plural packet-unit interleave packets PIP thus generated.

[0044]

The packeting portion 14 links a predetermined number of packet-unit interleave packets PIP that constitute the packet interleave data PID supplied from the packet interleaver 13 to generate one item of packet data PD. Specifically, to transmit a video signal and/or an audio signal in accordance with an RTP (Real-time Transport Protocol) known as a transfer protocol for streaming reproduction, the packeting portion 14 arbitrarily links the six consecutive packet-unit interleave packets PIP of a sequence of packet-unit interleave packets PIP and stores them in one packet of the RTP as an upper layer, thereby generating packet data PD.

[0045]

In this case, the six packet-unit interleave packets PIP linked as the packet data PD contains information of 12 bytes each, of 18x6=108 encoded packets EP. If an order of encoded packets EP is expressed as ..., 100, 101, ..., 136, ... as shown in FIG. 5(A), the encoder 10 disperses information contained in one encoded packet EP 12 bytes at a time into 18 consecutive byte interleave packets BIP as shown in FIG. 5(B). Furthermore, the encoder 10 aggregates one column of information in FIG. 5(C) to generate one packet-unit interleave packet PIP. Then, the encoder 10 uses the packeting portion 14 to link arbitrary six consecutive packet-unit interleave packets PIP, and thus generates one item of packet data PD as shown in FIG. 5(D). Therefore, each of the items of packet data PD contains information of 12 bytes each from 18x6=108 encoded packets EP as described above.

[0046]

Such packet data PD has no overlapping numbers as shown in FIG. 5(D), that is, constituted of information from 108 mutually-different encoded packets

EP. This is because, since 24 bytes of information is incorporated from one encoded packet EP if the same number is present in one item of packet data PD, in a case where one upper layer packet ULP is lost on the communication path, a situation where an encoded packet EP' having an error data quantity exceeding a maximum error correction capability of the decoder 20 occurs, thus making it impossible to completely correct errors. Therefore, the packeting portion 14 generates packet data PD by linking arbitrary six consecutive packet-unit interleave packets PIP in a sequence of packet-unit interleave packets PIP in such a manner that information from the same encoded packet EP is not incorporated. The packeting portion 14 only needs to link arbitrary six consecutive packet-unit interleave packets PIP irrespective of the position of a sink byte whose value has been inverted by the packet interleaver 13. The packeting portion 14 supplies the generated packet data PD to the subsequent transmission portion 15.

[0047]

For transmitting packet data PD supplied from the packeting portion 14 in accordance with the RTP and the TCP/IP (Transmission Control Protocol/Internet Protocol) using a so-called UDP (User Datagram Protocol) as a transport layer protocol, the transmission portion 15 adds an RTP/UDP/IP packet header to each of the items of packet data PD to generate one upper layer packet ULP, and transmits it as transmit data TD constituted of plural upper layer packets ULP.

[0048]

The encoder 10 having these portions outputs a transport stream TS input as an information series as transmit data TD, the transition of the packet format corresponding to the processing of each portion being shown in FIG. 6.

[0049]

That is, when a transport stream TS constituted of plural transport packets TSP shown on the first row in the figure is input, the encoder 10 performs Reed-Solomon encoding on each of these transport packets TSP by using the Reed-Solomon-encoding portion 11, and generates encoded data ED constituted of plural encoded packets EP shown on the second row in the figure. At this

time, it goes without saying that a quantity of information contained in one encoded packet EP indicated by a solid grid on the second row in the figure, more precisely, an information quantity of the encoded packet EP excluding an added error correcting code is the same as that of the transport packet TSP as indicated by the solid grid on the first row in the figure even when Reed-Solomon encoding is performed on the transport packet TSP by the Reed-Solomon-encoding portion 11, and an order of a series of encoded packets EP is also the same as that of a series of transport packets TSP.

[0050]

Then, as shown on the third row in the figure, the encoder 10 performs folding interleaving in units of a byte on each of the encoded packets EP by using the byte interleaver 12, to generate byte interleave data BID constituted of plural byte interleave packets BIP obtained by dispersing information contained in one encoded packet EP indicated by a solid grid on the second row in the figure 12 bytes at a time into 18 consecutive byte interleave packets BIP as indicated by a hatched portion shown on the third row in the figure. Note here that the hatched portions in the figure indicate a state where information quantity of one encoded packet EP indicated by the solid grid on the second row in the figure is dispersed into 18 consecutive byte interleave packets BIP.

[0051]

Further, the encoder 10 performs folding interleaving on each of the byte interleave packets BIP in units of a packet by the packet interleaver 13 as shown on the fourth and fifth rows in the figure, to generate packet interleave data PID constituted of plural packet-unit interleave packets PIP obtained by sporadically dispersing information contained in one encoded packet EP indicated by the solid grid on the second row in the figure into 144 consecutive packet-unit interleave packets PIP as indicated by the hatched portions on the fifth row in the figure, and in which information of 12 bytes each is dispersed with a spacing of at least six packet-unit interleave packets PIP therebetween.

[0052]

Then, as shown on the sixth row in the figure, the encoder 10 links six

consecutive packet-unit interleave packets PIP by using the packeting portion 14 to generate the packet data PD, and adds an RTP/UDP/IP packet header to each of the items of packet data PD and transmits it as transmit data TD constituted of plural upper layer packets ULP by the transmission portion 15.

[0053]

Thus, by performing folding interleaving in units of a byte and folding interleaving in units of a packet on the encoded packets EP to each of which a Reed-Solomon code has been added, the encoder 10 can send transmit data TD that has an error correction capability which enables a significant burst error containing a packet loss to be corrected using a Reed-Solomon code having a small code length. That is, even if a burst error has occurred in information contained in one upper layer packet ULP at maximum as in a case where one upper layer packet ULP is lost on the communication path, the encoder 10 can send transmit data TD that has an error correction capability which enables a burst error containing this packet loss to be corrected completely by the decoder 20.

[0054]

Further, using a fact that a value of beginning data of the packet is a fixed value, by inverting a value of a sink byte which is beginning data of a first packet-unit interleave packet PIP in performing folding interleaving in units of a packet, the encoder 10 can notify the reception side of the beginning packet in the packet interleave data PID.

[0055]

Furthermore, by the encoder 10 mounting the byte interleaver 12 using the DPRAM that has a small capacity and is incorporated into a programmable device and mounting the packet interleaver 13 using the SDRAM 16 that has a relatively large capacity and is externally attached to the programmable device, memory resources provided inside the programmable device can be saved and simplification of peripheral circuits due to burst access to the SDRAM 16 can be realized. Thus, by performing two-stage interleaving by use of the DPRAM and the SDRAM 16, the encoder 10 can complementarily avoid disadvantages of

these memories, and by using the externally attached SDRAM 16 instead of the small-capacity DPRAM incorporated in the programmable device when performing folding interleaving in units of a packet, costs can be cut and high-speed operations can be made by accessing the SDRAM 16 in units of a packet.

[0056]

Transmit data TD sent by such an encoder 10 is transmitted through a predetermined communication path and received by the decoder 20 as receive data RD constituted of plural upper layer packets ULP.

[0057]

Next, the decoder 20 in a data transmission/reception system will be described.

[0058]

As shown in FIG. 1 for example, the decoder 20 has a reception portion 21 for receiving and analyzing receive data RD constituted of plural upper layer packets ULP’, a packet consecutiveness verification portion 22 for verifying consecutiveness of upper layer packets ULP' on the basis of a header added to the receive data RD received by this reception portion 21, an un-packeting portion 23 for dividing packet data PD' received by the reception portion 21 and from which the header has been removed into packet-unit interleave packets PIP', a packet de-interleaver 24 for performing folding de-interleaving on the packet interleave data PID' in units of a packet in such a manner as to restore an order of the plural packet-unit interleave packets PIP' divided by this un-packeting portion 23 to an order of byte interleave packets BIP that constitute original byte interleave data BID, a byte de-interleaver 25 for performing folding de-interleaving on the byte interleave data BID' in units of a byte in such a manner as to restore an order of the plural byte interleave packets BIP' on which de-interleaving has been performed by this packet de-interleaver 24 to an order of the encoded packets EP that constitute the original encoded data ED, and a Reed-Solomon-decoding portion 26 for decoding a Reed-Solomon code of each of the encoded packets EP' constituting the encoded data ED' on which folding de-interleaving has been

performed by this byte de-interleaver 25 and outputting a transport stream TS constituted of plural transport packets TSP.

[0059]

In the decoder 20, of these portions, at least the packet consecutiveness verification portion 22, the un-packeting portion 23, an interface portion 28 in the packet de-interleaver 24, the byte de-interleaver 25, and the Reed-Solomon-decoding portion 26 which are enclosed by a broken line in the figure are mounted as a programmable device such as a PLD or an FPGA.

[0060]

The reception portion 21 is provided corresponding to the transmission portion 15 in the decoder 10 and analyzes an RTP/UDP/IP packet header added to each of the plural upper layer packets ULP' that constitute this receive data RD upon reception of the receive data RD through a predetermined communication path. Specifically, the reception portion 21 analyzes a sequence No. of an RTP in the RTP/UDP/IP packet header. The reception portion 21 supplies the packet consecutiveness verification portion 22 with analyzed information AN that indicates the sequence No. of the analyzed RTP and also supplies the subsequent un-packeting portion 23 with packet data PD' generated by removing the RTP/UDP/IP packet header from each of the upper layer packets ULP'.

[0061]

The packet consecutiveness verification portion 22 verifies, based on the analyzed information AN supplied from the reception portion 21, that is, the sequence No. of the RTP, consecutiveness of the upper layer packets ULP' regarding whether any of the upper layer packets ULP' is lost on the communication path. The packet consecutiveness verification portion 22 supplies verification result information CK that indicates a result of this verification to the interface portion 28 in the packet de-interleaver 24.

[0062]

The un-packeting portion 23 is provided corresponding to the packeting portion 14 in the encoder 10 and divides packet data PD' supplied from the reception portion 21 in such a manner as to restore packet data PD generated by

the packeting portion 14 to the packet-unit interleave packet PIP before linking, to thus generate packet-unit interleave packets PIP'. The un-packeting portion 23 supplies the subsequent packet de-interleaver 24 with packet interleave data PID' constituted of the plural packet-unit interleave packets PIP' thus generated.

[0063]

The packet de-interleaver 24 is provided corresponding to the packet interleaver 13 in the encoder 10 and performs folding interleaving with a depth of 6 and a period of 4 in units of byte interleave packets on the packet-unit interleave packets PIP' that constitute the packet interleave data PID' supplied from the un-packeting portion 23 in such a manner as to restore an order of packet interleave data PID on which folding interleaving has been performed by the packet interleaver 13 to an order of byte interleave packets BIP that constitute the original byte interleave data BID, to thus generate byte interleave data BID' constituted of the plural byte interleave packets BIP'. At this time, the packet de-interleaver 24 synchronizes, based on a sink byte whose value has been inverted by the packet interleaver 13, a beginning packet in the packet interleave data PID' supplied from the un-packeting portion 23 and performs folding de-interleaving in units of byte interleave packets.

[0064]

Specifically, the packet de-interleaver 24, like the packet interleaver 13, includes an SDRAM 27 which is externally attached to each of the portions mounted as a programmable device and the interface portion 28 for performing processing such as address generation and data transmission and reception for the purpose of performing data write and read with respect to this SDRAM 27. The packet de-interleaver 24 sequentially transfers and writes items of packet interleave data PID' input to the interface portion 28 to the SDRAM 27 in accordance with a predetermined write address and sequentially reads the data written to this SDRAM 27 in accordance with a predetermined read address that is different from the write address, thereby performing folding de-interleaving on the packet interleave data PID' supplied from the un-packeting portion 23.

[0065]

At this time, the packet de-interleaver 24 can cope with a case where items of the packet interleave data PID' are not consecutive, on the basis of verification result information CK supplied from the packet consecutiveness verification portion 22. For example, if it is grasped on the basis of the verification result information CK that one upper layer packet ULP' is lost on the communication path, the packet de-interleaver 24 reads data by performing an ordinary de-interleaving operation in accordance with a predetermined read address while writing data by skipping a region enough to store six packet-unit interleave packets PIP' that are tantamount to the one upper layer packet ULP' when writing data to the SDRAM 27. The data read from the SDRAM 27 by such an operation is data in which a portion corresponding to the lost packet is replaced with data read from the same read address in a previous time slot, that is, invalid data.

[0066]

The packet interleaver 24 restores a value “0xB8” of a sink byte inverted by the packet interleaver 13 out of the data obtained through such an operation to an original value of “0x47”, to generate byte interleave data BID' to be output eventually. The packet de-interleaver 24 supplies the subsequent byte de-interleaver 25 with the byte interleave data BID' constituted of the plural byte interleave packets BIP' thus generated. Note here that the packet interleaver 24 also has a function as a reception buffer to achieve a purpose of, for example, absorption of jitter occurring on the communication path, so the SDRAM 27 only needs to have a capacity to store at least 144 packet-unit interleave packets PIP' plus a little capacity to achieve the purpose.

[0067]

The byte de-interleaver 25 is provided corresponding to the byte interleaver 12 in the encoder 10 and performs folding de-interleaving with a depth of 18 and a period of 12 in units of a byte on each of the byte interleave packets BIP' that constitute the byte interleave data BID' supplied from the packet de-interleaver 24 in such a manner as to restore an order of the byte interleave data BID on which folding interleaving has been performed by the byte

interleaver 12 to an order of encoded packets EP that constitute an original encoded data ED, to thus generate encoded data ED' constituted of plural encoded packets EP'.

[0068]

Specifically, the byte de-interleaver 25, like the byte interleaver 12, is constituted of a DPRAM with which high-speed random access in units of a byte is possible, an address generation portion for performing data write and read with respect to this DPRAM, etc. The byte de-interleaver 25 sequentially transfers and writes items of input byte interleave data BID' to the DPRAM in accordance with a predetermined write address and sequentially reads the data written to the DPRAM in accordance with a predetermined read address that is different from the write address, thereby performing folding de-interleaving on the byte interleave data BID' supplied from the packet de-interleaver 24.

[0069]

Such a byte de-interleaver 25 supplies the subsequent Reed-Solomon-decoding portion 26 with encoded data ED' constituted of the plural encoded packet EP' thus generated. The encoded data ED' on which folding de-interleaving has been performed by this byte de-interleaver 25 becomes a state where 12-byte error data is contained in each of the 18x6=108 encoded packets EP' if, for example, one upper layer packet ULP' is lost on the communication path.

[0070]

Note here that like the byte interleaver 12, this byte de-interleaver 25 can control addresses according to a predetermined address generation system, to reduce a capacity of a memory required to perform folding de-interleaving in units of a byte, that is, the DPRAM, the descriptions of which will be given later.

[0071]

The Reed-Solomon-decoding portion 26 is provided corresponding to the Reed-Solomon-encoding portion 11 in the encoder 10 and performs decoding of a Reed-Solomon code on each of the encoded packets EP' that constitute the encoded data ED' supplied from the byte de-interleaver 25, to restore a transport

stream TS constituted of plural transport packets TSP. Specifically, the Reed-Solomon-decoding portion 26 performs error correction by using a 28-byte error correction code as a parity contained in each of the 216-byte encoded packets EP', to restore and output the transport stream TS constituted of 188-byte transport packets TSP. If, for example, one upper layer packet ULP' is lost on the communication path, this Reed-Solomon-decoding portion 26 can completely correct all bits since a maximum number of error bytes contained in an arbitrary encoded packet EP' is 12 bytes which is equal to or smaller than a maximum error correction bytes of 14 bytes.

[0072]

The decoder 20 having such portions completely restores the transport stream TS input as an information series to the encoder 10 from the receive data RD received through the predetermined communication path and outputs it, the transition of the packet format corresponding to the processing of each portion being shown in FIG. 7.

[0073]

That is, upon receiving the receive data RD through the reception portion 21 and generating packet data PD' shown on the first row in the figure, the decoder 20 divides these items of packet data PD' by using the un-packeting portion 23 and generates packet interleave data PID' constituted of plural packet-unit interleave packets PIP' shown on the second row in the figure. Here, it is assumed that the receive data RD is such that one upper layer packet ULP' is lost on the communication path and packet data PD' and a packet-unit interleave packet PIP' that correspond to this lost upper layer packet ULP' are those indicated by horizontal lines in the figure. At this time, information contained in one encoded packet EP generated by the encoder 10 is sporadically dispersed into 144 consecutive packet-unit interleave packets PIP' as indicated by hatched portions on the first and second rows in the figure, and information of 12 bytes each is dispersed with a spacing of at least six packet-unit interleave packets PIP' therebetween.

[0074]

The decoder 20 performs folding de-interleaving by using the packet de-interleaver 24 on each of the packet-unit interleave packets PIP' in units of a packet as shown on the third row in the figure, to generate byte interleave data BID' constituted of plural byte interleave packets BIP' obtained by dispersing, into 18 consecutive byte interleave packets BIP', 12 bytes at a time, information contained in one encoded packet EP generated by the encoder 10 as indicated by the hatched portion and the horizontal line on the third row in the figure. At this time, based on the verification result information CK of the packet consecutiveness verification portion 22, the decoder 20 replaces data that corresponds to the lost upper layer packet ULP' with invalid data, to generate byte interleave data BID' containing byte interleave packets BIP' indicated by the horizontal line having “?” attached to it on the third row in the figure.

[0075]

Furthermore, as shown on the fourth row in the figure, the decoder 20 performs folding interleaving in units of a byte on each of the byte interleave packets BIP' by using the byte de-interleaver 25, to generate encoded data ED' constituted of plural encoded packets EP'. It thus becomes possible to handle an extremely significant burst error as large as 216x6=1296 bytes owing to one lost upper layer packet ULP' as a 12-byte random error at most that is dispersed into the encoded packets EP' as indicated by horizontal lines on the fourth row in the figure.

[0076]

Then, as shown on the fifth row in the figure, the decoder 20 decodes a Reed-Solomon code by using the Reed-Solomon-decoding portion 26, to generate a transport stream TS constituted of plural transport packets TSP whose errors have been corrected completely. At this time, the dispersed information indicated by hatched portions in the figure is aggregated in one transport packet TSP as indicated by a solid grid on the fifth row in the figure.

[0077]

In such a manner, even if receive data RD that is in a state where a burst error has occurred in information contained in one upper layer packet ULP' at

most is received, the decoder 20 can completely correct this burst error. At this time, even if the receive data RD constituted of plural upper layer packets ULP' in which one of the upper layer packets ULP' in the transmit data TD generated by the encoder 10 has been lost on the communication path is received, since it can be regarded as a burst error due to the replacement of the lost absent data with invalid data, the decoder 20 can completely restore information that corresponds to the lost upper layer packet ULP'.

[0078]

Further, like the encoder 10, by the decoder 20 mounting the byte de-interleaver 25 by using the small-capacity DPRAM incorporated in the programmable device and mounting the packet de-interleaver 24 by using the SDRAM 27 that has a relatively large capacity and is externally attached to the programmable device, memory resources provided in the programmable device can be saved and simplification of peripheral circuits owing to burst access to the SDRAM 27 can be realized. By thus performing two-stage folding de-interleaving by use of the DPRAM and the SDRAM 27, like the encoder 10, the decoder 20 can complementarily avoid disadvantages of these memories, and by using the externally attached SDRAM 27 instead of the small-capacity DPRAM incorporated in the programmable device when performing folding de-interleaving in units of a packet, costs can be cut and high-speed operations can be made by accessing the SDRAM 27 in units of a packet.

[0079]

Note here that the byte interleaver 12 in the encoder 10 and the byte de-interleaver 25 in the decoder 20 can control addresses according to the predetermined address generation system as described above, to reduce the capacity of the memories required to perform folding interleaving and folding de-interleaving in units of a byte, the details of which will be described below.

[0080]

To perform folding interleaving or folding de-interleaving in units of a byte using the DPRAM as a memory, typically, data rearrangement for the folding interleaving or folding de-interleaving is performed on the DPRAM writing side,

whereas read is sequentially performed 1 packet at a time from the beginning address on the reading side. Conversely, the folding interleaving or folding de-interleaving in units of a byte can also be performed by sequentially performing write 1 packet at a time from the beginning address of the DPRAM on the writing side and rearranging data for the folding interleaving or folding de-interleaving on the DPRAM reading side. Therefore, in the DPRAM, even if there is a region from which data has been read because it has already been used to perform folding interleaving or folding de-interleaving, this region is not used immediately after the data is read and enters such a condition that about half a region reserved for the folding interleaving or folding de-interleaving is left unused.

[0081]

Therefore, when having read data from a certain region in the DPRAM, the byte interleaver 12 and the byte de-interleaver 25 each write the data to be written next to this region, to thus realize folding interleaving or folding de-interleaving. That is, the byte interleaver 12 and the byte de-interleaver 25 eliminate an already-used wasteful region out of the region in the DPRAM and constantly stores, in the preserved region, only significant data used thereafter for the folding interleaving or folding de-interleaving. Accordingly, the byte interleaver 12 and the byte de-interleaver 25 can each minimize the capacity of the DPRAM required to perform folding interleaving or folding de-interleaving.

[0082]

To realize data write and read operations with respect to such a DPRAM, the byte interleaver 12 and the byte de-interleaver 25 control addresses in accordance with an address generation system described with reference to FIGS. 8 to 15.

[0083]

Note here that hereinafter, to simplify descriptions, it is assumed that folding interleaving with a depth of 4 and a period of 3 is performed on a 12-byte encoded packet EP by the byte interleaver 12 to generate byte interleave packet BID of 12 bytes, and folding de-interleaving with a depth of 4 and a period of 3 is

performed on the byte interleave packet BID' by the byte de-interleaver 25 to generate 12-byte encoded packet EP'. That is, while the byte interleaver 12 and the byte de-interleaver 25 ordinarily need to be mounted using a memory including a capacity of the number of bytes represented by a product of the depth of the folding interleaving or folding de-interleaving and the number of bytes per input packet, specifically, 12x4=48 bytes, mounting is made possible by using memory having smaller capacity than that capacity.

[0084]

First, the byte interleaver 12 will be described.

[0085]

The byte interleaver 12 classifies bytes constituting a certain encoded packet EP in accordance with what packet delayed how much from the packet to which the byte belongs each of the bytes is stored due to the folding interleaving, and collectively manages the classified bytes in consecutive addresses on a memory map in the DPRAM. It is here assumed that the number of packets delayed from the packet to which a certain byte belongs is referred to as “delay” and a group of bytes having the same number of delay packets is referred to as “delay group”. That is, regarding the bytes constituting an arbitrary encoded packet EP, the byte interleaver 12 collectively classifies, as a delay group, the group of bytes having the same number of delay packets as the number of delays of the byte interleave packet BIP in which these bytes are stored due to the folding interleaving being performed on this encoded packet EP, and hands data for each delay group by generating a write address and a read address so that write and read are performed with respect to consecutive addresses in the DPRAM for each of the delay groups.

[0086]

To classify the bytes constituting a certain encoded packet EP into such a delay group, the byte interleaver 12 is provided with, as an address generation portion, two auxiliary counters, although not shown, for assisting in calculation of write and read addresses used to perform data write and read with respect to the DPRAM. Specifically, assuming that incoming 12-byte encoded packets EP are

each assigned with packet Nos. 0, 1, 2, 3, ... as shown in FIG. 8(A) and bytes constituting each of these encoded packets EP are assigned with byte Nos. 0 to 11 as shown in FIG. 8(B), the byte interleaver 12 is provided with, as auxiliary counters, a first counter for counting by incrementing count value cnt_del (0-3) that corresponds to a depth 4 of folding interleaving according to the byte No. as shown in FIG. 8(C) and a second counter for counting by incrementing count value cnt_block (0-2) that corresponds to a period of the first counter as shown in FIG. 8(D).

[0087]

Note here that count value cnt_del counted by the first counter indicates the above-mentioned number of delay packets, that is, a delay, while count value cnt_block counted by the second counter indicates the number of blocks assuming a period of the folding interleaving to be one block.

[0088]

The byte interleaver 12 uses such count value cnt_del counted by the first counter and count value cnt_block counted by the second counter, to control data write and read with respect to the DPRAM, thereby performing folding interleaving on the encoded packets EP. In this case, the byte interleaver 12, when having received encoded packets EP to undergo folding interleaving, starts writing data to the DPRAM immediately, while it starts reading data from the DPRAM simultaneously with data writing or as delayed by a predetermined fixed number of clock. When having received an encoded packet EP, to avoid such a situation that significant data written to the DPRAM may be overwritten by any other data before the significant data is read, the byte interleaver 12 needs to read all the data completely before the next encoded packet EP is input after the data is written to the DPRAM.

[0089]

Specifically, the byte interleaver 12 performs write and read of byte groups classified into delay groups to the DPRAM as shown in, for example, FIGS. 9(A) to 9(D). It is supposed here that a group of bytes constituted of three bytes in accordance with count value cnt_del of “0” is referred to as delay group

del0, a group of bytes constituted of three bytes in accordance with count value cnt_del of “1” is referred to as delay group del1, a group of bytes constituted of three bytes in accordance with count value cnt_del of “2” is referred to as delay group del2, and a group of bytes constituted of three bytes in accordance with count value cnt_del of “3” is referred to as delay group del3. Further, in FIGS. 9(A) to 9(D), delay groups del0, del1, del2, and del3 in an encoded packet EP with a packet No. of “n” are supposed to be written as “n-0”, “n-1”, “n-2”, and “n-3” respectively.

[0090]

That is, in preparation for performing folding interleaving, the byte interleaver 12 sets to a predetermined initial value such as “0” an reference address addr_base that is an address of the DPRAM at a moment when the first counter and the second counter are released from a reset state and that indicates a beginning from which data write and read with respect to the DPRAM start as indicated by a bold horizontal line in the leftmost portion in FIG. 9(B). Then, when having received an encoded packet EP having packet No. “0” as indicated by the leftmost packet No. in FIG. 9(A), it classifies bytes constituting this encoded packet EP into four three-byte delay groups of del0 (0-0), del1 (0-1), del2 (0-2), and del3 (0-3) on the basis of count value cnt_del given by the first counter as indicated by the leftmost portion in FIG. 9(B).

[0091]

Further, the byte interleaver 12 generates a write address in such a manner that, as indicated by the leftmost portion in FIG. 9(C), of these four delay groups del0 (0-0), del1 (0-1), del2 (0-2), and del3 (0-3), the bytes belonging to delay group del1 (0-0) are not written to the DPRAM, the bytes belonging to delay group del1 (0-1) are written to region w1 which is two blocks behind the above-mentioned reference address addr_base in the DPRAM, the bytes belonging to delay group del2 (0-2) are written to region w2 which is (2+3) blocks behind the above-mentioned reference address addr_base in the DPRAM, and the bytes belonging to delay group del3 (0-3) are written to a region which is (2+3+4) blocks behind the above-mentioned reference address addr_base in the

DPRAM, that is, the circulated beginning region w3.

[0092]

Further, the byte interleaver 12 performs a read operation as well as such a write operation with respect to the DPRAM. That is, the byte interleaver 12 generates a read address in such a manner that, of four delay groups del0, del1, del2, and del3 written in the DPRAM, the bytes belonging to delay group del0 are read as they are because these bytes are not written to the DPRAM, the bytes belonging to delay group del1 are read from region r1 which is one block behind the reference address addr_base in the DPRAM, the bytes belonging to delay group del2 are read from region r2 which is (1+2) blocks behind the reference address addr_base in the DPRAM, and the bytes belonging to delay group del3 are read from region r3 which is (1+2+3) blocks behind the reference address addr_base in the DPRAM. However, since no data is written in either of regions r1, r2, or r3 at this moment, there is actually no data to be read by the byte interleaver 12.

[0093]

Therefore, the byte interleaver 12 outputs data constituted of the bytes classified as delay group del0 (0-0) as indicated by the leftmost portion in FIG. 9(D).

[0094]

Subsequently, in preparation for inputting the next encoded packet EP, the byte interleaver 12 moves forward the reference address addr_base by one block as indicated by the bold horizontal line in the second leftmost portion in FIG. 9(B) and, as indicated by the second leftmost portion in FIG. 9(A), inputs the next encoded packet EP with packet No. "1" to thereby, as indicated by the second leftmost portion in FIG. 9(B), classify the bytes that constitute this encoded packet EP into four three-byte delay groups del0 (1-0), del1 (1-1), del2 (1-2), and del3 (1-3) on the basis of count value cnt_del given by the first counter.

[0095]

Then, the byte interleaver 12, as indicated by the second leftmost portion in FIG. 9(C), generates a write address in such a manner that, of these four delay

groups del0 (1-0), del1 (1-1), del2 (1-2), and del3 (1-3), the bytes belonging to delay group del0 (1-0) are not written to the DPRAM, the bytes belonging to delay groups del1 (1-1) are written to region w1 which is two blocks behind the reference address addr_base in the DPRAM, the bytes belonging to delay group del2 (1-2) are written to region w2 which is (2+3) blocks behind the reference address addr_base in the DPRAM, and the bytes belonging to delay group del3 (1-3) are written to the region which is (2+3+4) blocks behind the reference address addr_base in the DPRAM, that is, the circulated beginning region w3.

[0096]

Further, the byte interleaver 12 generates a read address in such a manner that, of four delay groups del0, del1, del2, and del3 written to the DPRAM, the bytes belonging to delay group del0 are read as they are because they are not written to the DPRAM, the bytes belonging to delay group del1 are read from region r1 which is one block behind the reference address addr_base in the DPRAM, the bytes belonging to delay group del2 are read from region r2 which is (1+2) blocks behind the reference address addr_base in the DPRAM, and the bytes belonging to delay group del3 are read from region r3 which is (1+2+3) blocks behind the reference address addr_base in the DPRAM. Note here that the byte interleaver 12 actually reads only the delay group del1 (0-1) because no data is written to regions r2 and r3 and region r1 is the same as region w1 at the time when an encoded packet EP with packet No. "0" is input, in which delay group del1 (0-1) is written.

[0097]

Accordingly, the byte interleaver 12, as indicated by the second leftmost portion in FIG. 9(D), outputs data constituted of the bytes classified as delay groups del0 (1-0) and del1 (0-1).

[0098]

Subsequently, in preparation for inputting the next encoded packet EP, the byte interleaver 12 moves forward the reference address addr_base by another one block as indicated by the bold horizontal line in the third leftmost portion in FIG. 9(B) and, as indicated by the third leftmost portion in FIG. 9(A), inputs the

next encoded packet with packet No. “2” and, as indicated by the third leftmost portion in FIG. 9(B), classifies the bytes that constitute this encoded packet EP into four three-byte delay groups del0 (2-0), del1 (2-1), del2 (2-2), and del3 (2-3) on the basis of count value cnt_del given by the first counter.

[0099]

Then, the byte interleaver 12 generates a write address in such a manner that, as indicated by the third leftmost portion in FIG. 9(C), of these four delay groups del0 (2-0), del1 (2-1), del2 (2-2), and del3 (2-3), the bytes belonging to delay group del1 (2-0) are not written to the DPRAM, the bytes belonging to delay group del1 (2-1) are written to region w1 which is two blocks behind the reference address addr_base in the DPRAM, the bytes belonging to delay group del2 (2-2) are written to region w2 which is (2+3) blocks behind the reference address addr_base in the DPRAM, and the bytes belonging to delay group del3 (2-3) are written to a region which is (2+3+4) blocks behind the reference address addr_base in the DPRAM, that is, the circulated beginning region w3.

[0100]

Further, the byte interleaver 12 generates a read address in such a manner that, of four delay groups del0, del1, del2, and del3 written to the DPRAM, the bytes belonging to delay group del0 are read as they are because these bytes are not written to the DPRAM, the bytes belonging to delay group del1 are read from region r1 which is one block behind the reference address addr_base in the DPRAM, the bytes belonging to delay group del2 are read from region r2 which is (1+2) blocks behind the reference address addr_base in the DPRAM, and the bytes belonging to delay group del3 are read from region r3 which is (1+2+3) blocks behind the reference address addr_base in the DPRAM. At this moment, though no data is written in regions r3, since region r1 is the same as region w1 at the time when an encoded packet with packet No. “1” is input, and region r2 is the same as region w2 at the time when an encoded packet EP with packet No. “0” is input and has delay groups del1 (1-1) and del2 (0-2) written in it, the byte interleaver 12 actually reads only these delay groups del1 (1-1) and del2 (0-2).

[0101]

Accordingly, the byte interleaver 12 outputs data constituted of the bytes classified as delay groups del0 (2-0), del1 (1-1), and del2 (0-2) as indicated by the third leftmost portion in FIG. 9(D).

[0102]

Similarly, in preparation for inputting the next encoded packet EP, the byte interleaver 12 moves forward the reference address addr_base by another one block as indicated by the bold horizontal line in the fourth leftmost portion in FIG. 9(B) and, as indicated by the fourth leftmost portion in FIG. 9(A), inputs the next encoded packet EP with packet No. "3" to thereby, as indicated by the fourth leftmost portion in FIG. 9(B), classify the bytes that constitute this encoded packet EP into four three-byte delay groups del0 (3-0), del1 (3-1), del2 (3-2), and del3 (3-3) on the basis of count value cnt_del given by the first counter.

[0103]

Then, the byte interleaver 12 generates a write address in such a manner that, as indicated by the fourth leftmost portion in FIG. 9(C), of these four delay groups del0 (3-0), del1 (3-1), del2 (3-2), and del3 (3-3), the bytes belonging to delay group del0 (3-0) are not written to the DPRAM, the bytes belonging to delay groups del1 (3-1) are written to region w1 which is two blocks behind the reference address addr_base in the DPRAM, the bytes belonging to delay group del2 (3-2) are written to region w2 which is (2+3) blocks behind the reference address addr_base in the DPRAM, and the bytes belonging to delay group del3 (3-3) are written to a region which is (2+3+4) blocks behind the reference address addr_base in the DPRAM, that is, the circulated beginning region w3.

[0104]

It should be noted that, at this moment, details of the data mapped to the addresses in the DPRAM are such as shown in FIG. 10. That is, in the byte interleaver 12, as for the regions that can store three-byte blocks of from address 0 to address 2, delay group del3 (0-3) which has packet No. "0" and the number of delay packets "3" is stored, in which the bytes belonging to the delay groups are written with respect to the consecutive addresses starting from those having the lower numbered byte Nos. in such a manner that the data with byte No. "3"

may be stored in a region of address 0, the data with byte No. "7" may be stored in a region of address 1, and the data with byte No. "11" may be stored in a region of address 2.

[0105]

Further, the byte interleaver 12 generates a read address in such a manner that, as indicated by the fourth leftmost portion in FIG. 9(C), of four delay groups del0, del1, del2, and del3 written to the DPRAM, the bytes belonging to delay group del0 are read as they are because they are not written to the DPRAM, the bytes belonging to delay group del1 are read from region r1 which is one block behind the reference address addr_base in the DPRAM, the bytes belonging to delay group del2 are read from region r2 which is (1+2) blocks behind the reference address addr_base in the DPRAM, and the bytes belonging to delay group del3 are read from region r3 which is (1+2+3) blocks behind the reference address addr_base in the DPRAM. In this case, region r1 is the same as region w1 at the time when an encoded packet EP with packet No. "2" is input, region r2 is the same as region w2 at the time when an encoded packet EP with packet No. "1" is input, and region r3 is the same as region w3 at the time when an encoded packet EP with packet No. "0" is input and has delay groups del1 (2-1), del2 (1-2), and del3 (0-3) written in it, so that the byte interleaver 12 actually reads these delay groups del1 (2-1), del2 (1-2), and del3 (0-3).

[0106]

Accordingly, the byte interleaver 12 outputs data constituted of the bytes classified as delay groups del0 (3-0), del1 (2-1), del2 (1-2), and del3 (0-3) as indicated by the fourth leftmost portion in FIG. 9(D).

[0107]

In such a manner, the byte interleaver 12 classifies the bytes into the delay groups each time an encoded packet EP is input and, based on write address and read address in accordance with the above-mentioned address generation system, performs managed write and read with respect to the consecutive addresses on a memory map in the DPRAM for each of the delay groups. With this, since the byte interleaver 12 can immediately write the next data to the

region of the DPRAM from which data is read, it is possible to eliminate a useless region, which is considered to be the used, of the DPRAM regions, thus suppressing a required capacity of the DPRAM to the minimum. In this case, although typically the byte interleaver 12 requires such a DPRAM as to have a capacity equal to a product of a depth D=4 of folding interleaving and the number of bytes N=12 per input packet, that is, DxN=4x12=48 bytes, it only need to use such a DPRAM as to have a capacity of only nine three-byte regions, that is, 27 bytes as shown in FIG. 9(C) by performing the above-mentioned address control, to thereby perform the similar folding interleaving. Further, the byte interleaver 12 can perform a write to consecutive addresses for each of the classified delay groups and a read for each of the delay groups written to these consecutive addresses, thereby easily calculating a write address and a read address.

[0108]

Next, the byte de-interleaver 25 will be described.

[0109]

The byte de-interleaver 25, like the byte interleaver 12, classifies bytes that constitute a byte interleave packet BIP' into delay groups and handles data for each of these delay groups, to perform address control symmetrical to the byte interleaver 12, thereby performing folding de-interleaving.

[0110]

That is, the byte de-interleaver 25, like the byte interleaver 12, is provided with, as an address generation portion, two auxiliary counters, although not shown, for assisting in calculation of addresses used to perform data write and read with respect to the DPRAM. Specifically, assuming that incoming byte interleave packets BIP' constituted of 12 bytes are each assigned with packet Nos. 0, 1, 2, 3, ... as shown in FIG. 11(A) and bytes of each of these byte interleave packets BIP' are assigned with byte Nos. 0 to 11 as shown in FIG. 11(B), the byte de-interleaver 25 is provided with, as auxiliary counters, a first counter for counting by decrementing count value cnt_del (0-3) that corresponds to a depth 4 of folding de-interleaving according to the byte No. as shown in FIG. 11(C) and a second counter for counting by incrementing count value cnt_block (0-2) that

corresponds to a period of the first counter as shown in FIG. 11(D). Note here that a difference from the byte interleaver 12 is in that count value cnt_del counted by the first counter is not incremented but decremented.

[0111]

Like the byte interleaver 12, the byte de-interleaver 25 uses such count value cnt_del counted by the first counter and count value cnt_block counted by the second counter, to control data write and read with respect to the DPRAM, thereby performing folding de-interleaving on the byte interleave packets BIP'. The byte de-interleaver 25, when having received byte interleave packets BIP' to undergo folding de-interleaving, starts writing data to the DPRAM immediately, while it starts reading data from the DPRAM simultaneously with data writing or as delayed by a predetermined fixed number of clocks. When having received a byte interleave packet BIP', to avoid such a situation that significant data written to the DPRAM may be overwritten by any other data before the significant data is read, the byte de-interleaver 25 needs to read all the data completely before the next byte interleave packet BIP' is input after the data is written to the DPRAM.

[0112]

Specifically, when having received byte interleave packets BIP' constituted of a data array output from the byte interleaver 12, the byte de-interleaver 25 performs write and read of byte groups classified into delay groups with respect to the DPRAM as shown in, for example, FIGS. 12. It is supposed here, as in the case shown in FIGS. 9 earlier, that a group of bytes constituted of three bytes in accordance with count value cnt_del of "0" is referred to as delay group del0, a group of bytes constituted of three bytes in accordance with count value cnt_del of "1" is referred to as delay group del1, a group of bytes constituted of three bytes in accordance with count value cnt_del of "2" is referred to as delay group del2, and a group of bytes constituted of three bytes in accordance with count value cnt_del of "3" is referred to as delay group del3.

[0113]

That is, in preparation for performing folding de-interleaving, the byte

de-interleaver 25 sets to a predetermined initial value such as "0" the reference address addr_base that is an address of the DPRAM at a moment when the first counter and the second counter are released from a reset state and that indicates a beginning from which data write and read with respect to the DPRAM start as indicated by the bold horizontal line in the leftmost portion in FIG. 12(B), and when having received a byte interleave packet BIP' having packet No. "0" as indicated by the leftmost packet portion in FIG. 12(A), classifies bytes constituting this byte interleave packet BIP' into four three-byte delay groups of del0 (0-3), del1 (1-2), del2 (2-1), and del3 (3-0) based on count value cnt_del given by the first counter as indicated by the leftmost portion in FIG. 12(B).

[0114]

Note here that in contrast to FIGS. 9 given earlier, in FIG. 12(B), count value cnt_del given by the first counter and a value of "m" of a delay group written as "n-m" are made different from each other for the purpose of easily explaining how to restore an encoded packet EP' that corresponds to an encoded packet EP by performing folding interleaving on the byte interleave packet BIP'.

[0115]

Then, the byte interleaver 25 generates a write address in such a manner that, as indicated by the leftmost portion in FIG. 12(C), of these four delay groups del0 (0-3), del1 (1-2), del2 (2-1), and del3 (3-0), the bytes belonging to delay group del0 (0-3) are not written to the DPRAM, the bytes belonging to delay group del1 (1-2) are written to region w3 which is two blocks behind the above-mentioned reference address addr_base in the DPRAM, the bytes belonging to delay group del2 (2-1) are written to region w2 which is (2+3) blocks behind the reference address addr_base in the DPRAM, the bytes belonging to delay group del3 (3-0) are written to a region which is (2+3+4) blocks behind the reference address addr_base in the DPRAM, that is, the circulated beginning region w1.

[0116]

Further, the byte de-interleaver 25 performs a read operation as well as such a write operation to the DPRAM. That is, the byte de-interleaver 25

generates a read address in such a manner that, of four delay groups del0, del1, del2, and del3 written in the DPRAM, the bytes belonging to delay group del0 are read as they are because these bytes are not written to the DPRAM, the bytes belonging to delay group del1 are read from region r3 which is one block behind the reference address addr_base in the DPRAM, the bytes belonging to delay group del2 are read from region r2 which is (1+2) blocks behind the reference address addr_base in the DPRAM, and the bytes belonging to delay group del3 are read from region r1 which is (1+2+3) blocks behind the reference address addr_base in the DPRAM. However, since no data is written in either of regions r1, r2, or r3 at this moment, there is actually no data to be read by the byte de-interleaver 25.

[0117]

Therefore, the byte de-interleaver 25 outputs data constituted of the bytes classified as delay group del0 (0-3) as indicated by the leftmost portion in FIG. 12(D).

[0118]

Subsequently, in preparation for inputting the next byte interleave packet BIP', the byte de-interleaver 25 moves forward the reference address addr_base by one block as indicated by the bold horizontal line in the second leftmost portion in FIG. 12(B) and, as indicated by the second leftmost portion in FIG. 12(A), inputs the next byte interleave packet BIP' with packet No. “1” to thereby, as indicated by the second leftmost portion in FIG. 12(B), classifies the bytes that constitute this byte interleave packet BIP' into four three-byte delay groups del0 (1-3), del1 (2-2), del2 (3-1), and del3 (0-0) on the basis of count value cnt_del given by the first counter.

[0119]

Then, the byte de-interleaver 25, as indicated by the second leftmost portion in FIG. 12(C), generates a write address in such a manner that, of these four delay groups del0 (1-3), del1 (2-2), del2 (3-1), and del3 (0-0), the bytes belonging to delay group del0 (1-3) are not written to the DPRAM, the bytes belonging to delay group del1 (2-2) are written to region w3 which is two blocks

behind the reference addr_base in the DPRAM, the bytes belonging to delay group del2 (3-1) are written to region w2 which is (2+3) blocks behind the reference address addr_base in the DPRAM, and the bytes belonging to delay group del3 (0-0) are written to the region which is (2+3+4) blocks behind the reference address addr_base in the DPRAM, that is, the circulated beginning region w1.

[0120]

Further, the byte de-interleaver 25 generates a read address in such a manner that, of four delay groups del0, del1, del2, and del3 written to the DPRAM, the bytes belonging to group del0 are read as they are because they are not written to the DPRAM, the bytes belonging to delay group del1 are read from region r3 which is one block behind the reference address addr_base in the DPRAM, the bytes belonging to delay group del2 are read from region r2 which is (1+2) blocks behind the reference address addr_base in the DPRAM, and the bytes belonging to delay group del3 are read from region r1 which is (1+2+3) blocks behind the reference address addr_base in the DPRAM. Note here that the byte de-interleaver 25 actually reads only the delay group del1 (1-2) because no data is written to regions r1 and r2 and region r3 is the same as region w3 at the time when a byte interleave packet BIP' with packet No. “0” is input, in which the delay group del1 (1-2) is written.

[0121]

Accordingly, the byte de-interleaver 25, as indicated by the second leftmost portion in FIG. 12(D), outputs data constituted of the bytes classified as delay groups del0 (1-3) and del1 (1-2).

[0122]

Subsequently, in preparation for inputting the next byte interleave packet BIP', the byte de-interleaver 25 moves forward the reference address addr_base by another one block as indicated by the bold horizontal line in the third leftmost portion in FIG. 12(B) and, as indicated by the third leftmost portion in FIG. 12(A), inputs the next byte interleave packet BIP' with packet No. “2” to thereby, as indicated by the third leftmost portion in FIG. 12(B), classify the bytes that

constitute this byte interleave packet BIP' into four three-byte delay groups del0 (2-3), del1 (3-2), del2 (0-1), and del3 (1-0) on the basis of count value cnt_del given by the first counter.

[0123]

Then, the byte de-interleaver 25 generates a write address in such a manner that, as indicated by the third leftmost portion in FIG. 12(C), of these four delay groups del0 (2-3), del1 (3-2), del2 (0-1), and del3 (1-0), the bytes belonging to delay group del0 (2-3) are not written to the DPRAM, the bytes belonging to delay group del1 (3-2) are written to region w3 which is two blocks behind the reference address addr_base in the DPRAM, the bytes belonging to delay group del2 (0-1) are written to region w2 which is (2+3) blocks behind the reference address addr_base in the DPRAM, the bytes belonging to delay group del3 (1-0) are written to a region which is (2+3+4) blocks behind the reference address addr_base in the DPRAM, that is, the circulated beginning region w1.

[0124]

Further, the byte de-interleaver 25 generates a read address in such a manner that, of four delay groups del0, del1, del2, and del3 written in the DPRAM, the bytes belonging to delay group del0 are read as they are because these bytes are not written to the DPRAM, the bytes belonging to delay group del1 are read from region r3 which is one block behind the reference address addr_base in the DPRAM, the bytes belonging to delay group del2 are read from region r2 which is (1+2) blocks behind the reference address addr_base in the DPRAM, and the bytes belonging to delay group del3 are read from region r1 which is (1+2+3) blocks behind the reference address addr_base in the DPRAM. At this moment, since no data is written in region r1, region r2 is the same as region w2 at the time when a byte interleave packet BIP' with packet No. “0” is input, and region r3 is the same as region w3 at the time when a byte interleave packet BIP’ with packet No. “1” is input, in which delay groups del2 (2-1) and del1 (2-2) are written, the byte de-interleaver 25 actually reads only these delay groups del2 (2-1) and del1 (2-2).

[0125]

Accordingly, the byte de-interleaver 25 outputs data constituted of the bytes classified as delay groups del0 (2-3), del1 (2-2), and del2 (2-1) as indicated by the third leftmost portion in FIG. 12(D).

[0126]

Similarly, in preparation for inputting the next byte interleave packet BIP', the byte de-interleaver 25 moves forward the reference address addr_base by another one block as indicated by the bold horizontal line in the fourth leftmost portion in FIG. 12(B) and, as indicated by the fourth leftmost portion in FIG. 12(A), inputs the next byte interleave packet BIP' with packet No. "3" to thereby, as indicated by the fourth leftmost portion in FIG. 12(B), classify the bytes that constitute this byte interleave packet BIP' into four three-byte groups del0 (3-3), del1 (0-2), del2 (1-1), and del3 (2-0) on the basis of count value cnt_del given by the first counter.

[0127]

Then, the byte de-interleaver 25 generates a write address in such a manner that, as indicated by the fourth leftmost portion in FIG. 12(C), of these four delay groups del0 (3-3), del1 (0-2), del2 (1-1), and del3 (2-0), the bytes belonging to delay group del0 (3-3) are not written to the DPRAM, the bytes belonging to delay group del1 (0-2) are written to region w3 which is two blocks behind the reference addr_base in the DPRAM, the bytes belonging to delay group del2 (1-1) are written to region w2 which is (2+3) blocks behind the reference address addr_base in the DPRAM, and the bytes belonging to delay group del3 (2-0) are written to a region which is (2+3+4) blocks behind the reference address addr_base in the DPRAM, that is, the circulated beginning region w1.

[0128]

At this moment, details of the data mapped to the addresses in the DPRAM are such as shown in FIG. 13. That is, in the byte de-interleaver 25, as for the regions that can store three-byte blocks of from address 0 to address 2, delay group del3 (3-0) which has packet No. "3" and the number of delay packets "3" is stored, in which the bytes belonging to the delay groups are written to the

consecutive addresses starting from those having the lower numbered byte Nos. in such a manner that the data with byte No. “0” may be stored in a region of address 0, the data with byte No. “4” may be stored in a region of address 1, and the data with byte No. “81” may be stored in a region of address 2.

[0129]

Further, the byte de-interleaver 25 generates a read address in such a manner as to, as indicated by the fourth leftmost portion in FIG. 12(C), of the four delay groups of del0, del1, del2, and del3 which are written to DPRAM, read the bytes belonging to delay group del0 as they are because they are not written to the DPRAM, the bytes belonging to delay group del1 from region r3 which is one block behind the reference address addr_base in the DPRAM, the bytes belonging to delay group del2 from region r2 which is (1+2) blocks behind the reference address addr_base in the DPRAM, and the bytes belonging to delay group del3 from region r1 which is (1+2+3) blocks behind the reference address addr_base in the DPRAM. In this case, region r1 is the same as region w1 at the time when a byte interleave packet BIP' with packet No. “0” is input, region r2 is the same as region w2 at the time when a byte interleave packet BIP' with packet No. “1” is input, and region r3 is the same as region w3 at the time when a byte interleave packet BIP' with packet No. “2” is input and has delay groups del3 (3-0), del2 (3-1), and del1 (3-2) written in it, so that the byte de-interleaver 25 actually reads these delay groups del3 (3-0), del2 (3-1), and del1 (3-2).

[0130]

Accordingly, the byte de-interleaver 25 outputs data constituted of the bytes classified as delay groups del0 (3-3), del1 (3-2), del2 (3-1), and del3 (3-0) as indicated by the fourth leftmost portion in FIG. 12(D).

[0131]

In such a manner, the byte de-interleaver 25 classifies the bytes into the delay groups each time a byte interleave packet BIP' is input and, based on write address and read address in accordance with the above-mentioned address generation system, performs managed write and read with respect to the consecutive addresses on a memory map in the DPRAM for each of the delay

groups. With this, like the byte interleaver 12, the byte de-interleaver 25 can immediately write the next data to the region of the DPRAM from which data is read, and therefore it is possible to eliminate a useless region, which is considered to be the used, of the DPRAM regions, thus suppressing a required capacity of the DPRAM to the minimum. In this case, typically the byte de-interleaver 25 requires such a DPRAM as to have a capacity equal to a product of a depth D=4 of folding de-interleaving and the number of bytes N=12 per input packet, that is, DxN=4x12=48 bytes, but only needs to use such a DPRAM as to have a capacity of only nine three-byte regions, that is, 27 bytes as shown in FIG. 12(C) by performing the above-mentioned address control, to perform the similar folding de-interleaving. Further, the byte de-interleaver 25 performs write to consecutive addresses for each of the classified delay groups and read from these consecutive addresses for each of the written delay groups, thereby being easily capable of calculating a write address and a read address.

[0132]

As described above, in a case of performing folding interleaving or folding de-interleaving with a depth of 4 and a period of 3 by inputting 12-byte packets in units of a byte, the byte interleaver 12 and the byte de-interleaver 25 may handle the data for each delay group by having a first counter for counting count value cnt_del (0-3) that corresponds to a depth 4 of folding interleaving or folding de-interleaving according to a byte No. of each of the bytes that constitute an incoming packet and a second counter for counting count value cnt_block (0-2) that corresponds to a period of this first counter.

[0133]

Therefore, assuming that the number of bytes per incoming packet to be N, a depth of folding interleaving or folding de-interleaving to be D, and a period of folding interleaving or folding de-interleaving to be C, the byte interleaver 12 and the byte de-interleaver 25 may perform data write and read with respect to the DPRAM for each delay group by having a first counter for counting count value cnt_del (0 through D-1) that corresponds to depth D of folding interleaving or folding de-interleaving according to a byte No. of each of the bytes that constitute

an incoming packet and a second counter for counting count value cnt_block (0 through C-1) that corresponds to a period of this first counter. That is, the address generation system by the byte interleaver 12 and the byte de-interleaver 25 can be generalized as follows.

[0134]

First, the byte interleaver 12 and the byte de-interleaver 25, to write data to the DPRAM, each generate a write address that corresponds to each delay group according to rules indicated by a series of processing steps shown in FIG. 14, thereby controlling data writing to the DPRAM.

[0135]

That is, as shown in FIG. 14, the byte interleaver 12 and the byte de-interleaver 25 each set the above-mentioned reference address addr_base to "0" at step S1 and, at step S2, judge whether there is an input packet to undergo folding interleaving or folding de-interleaving.

[0136]

If it is judged that there is no input packet, the byte interleaver 12 and the byte de-interleaver 25 each operate the first counter and the second counter to write the bytes for each delay group to the DPRAM on the basis of count value cnt_del given by the first counter.

[0137]

That is, the byte interleaver 12 and the byte de-interleaver 25, if count value cnt_del by the first counter 1 is set to "0" at step S3, each classify the current byte into delay group del0 and, at step S4, generate no write address and go to processing of step S2 without writing the bytes belonging to delay group del0 to the DPRAM.

[0138]

Further, the byte interleaver 12 and the byte de-interleaver 25, if count value cnt_del by the first counter is set to "1" at step S3, each classify the current byte into delay group del1 and, at step S5, generate the following:

(addr_base+Cx2+cnt_block)

as a write address to write the bytes belonging to delay group del1 to a region

indicated by this write address and go to processing of step S2.

[0139]

Furthermore, if count value cnt_del by the first counter is set to “2” at step S3, the byte interleaver 12 and the byte de-interleaver 25 each classify the current byte into delay group del2 and, at step S6, generate the following:

(addr_base+Cx(2+3)+cnt_block)

as a write address to write the bytes belonging to delay group del2 del1 to a region indicated by this write address and go to processing of step S2.

[0140]

Furthermore, if count value cnt_del by the first counter is set to “3” at step S3, the byte interleaver 12 and the byte de-interleaver 25 each classify the current byte into delay group del3 and, at step S7, generate the following:

(addr_base+Cx(2+3+4)+cnt_block)

as a write address to write the bytes belonging to delay group del3 to a region indicated by this write address and go to processing of step S2.

[0141]

If count value cnt_del by the first counter is set to “D-1” at step S3, the byte interleaver 12 and the byte de-interleaver 25 each classify the current byte into delay group del (D-1) and, at step S8, generate the following:

(addr_base+Cx(2+3+ …+D)+cnt_block)

as a write address to write the bytes belonging to delay group del (D-1) to a region indicated by this write address and go to processing of step S2.

[0142]

The byte interleaver 12 performs such an operation by incrementing count value cnt_del from “0” to “D-1” until count value cnt_del becomes “D-1” and repeats this operation as many times as a value of the period C of folding interleaving, to end the processing on all of the bytes included in each packet. The byte de-interleaver 25, on the other hand, performs such an operation by decrementing count value cnt_del from “D-1” to “0” until count value cnt_del becomes “0” and repeats this operation as many times as the value of the period C of the folding de-interleaving, to end the processing on all of the bytes included in

each packet.

[0143]

In such a manner, the byte interleaver 12 and the byte de-interleaver 25 each end the processing on the input packet and, if it has judged at step S2 that there is no input packet, go to process of step S9. Then, the byte interleaver 12 and the byte de-interleaver 25 respectively increase only once the reference address addr_base by a value of period C of folding interleaving or folding de-interleaving, that is, by one block, at step S9 in the wake of stopping incoming packet and go to processing of step S2 again to wait for the next incoming packet.

[0144]

The byte interleaver 12 and the byte de-interleaver 25 can each generate a write address for each delay group in accordance with such rules, to control data writing to the DPRAM.

[0145]

To read data from the DPRAM, on the other hand, the byte interleaver 12 and the byte de-interleaver 25 each generate a read address that corresponds to each delay group according to rules indicated by a series of processing shown in FIG. 15, thereby controlling data read from the DPRAM.

[0146]

That is, as shown in FIG. 15, the byte interleaver 12 and the byte de-interleaver 25 each set the above-mentioned reference address addr_base to "0" at step S11 and, at step S12, judge whether there is an input packet to undergo folding interleaving or folding de-interleaving.

[0147]

If it is judged here that there is an input packet, the byte interleaver 12 and the byte de-interleaver 25 each operate the first counter and the second counter to read the bytes for each delay group from the DPRAM on the basis of count value cnt_del given by the first counter.

[0148]

That is, the byte interleaver 12 and the byte de-interleaver 25, if count value cnt_del by the first counter 1 is set to "0" at step S13, each classify the

current byte into delay group del0 and, at step S14, generate no read address and output the bytes belonging to delay group del0 as they are without writing to the DPRAM, then going to processing of step S12.

[0149]

Further, the byte interleaver 12 and the byte de-interleaver 25, if count value cnt_del by the first counter is set to "1" at step S13, each generate the following:

(addr_base+Cx1+cnt_block)

as a write address at step S15 to read as delay group del1 the bytes stored in a region indicated by this read address and go to processing of step S12.

[0150]

Furthermore, if count value cnt_del by the first counter is set to "2" at step S13, the byte interleaver 12 and the byte de-interleaver 25 each generate the following:

(addr_base+Cx(1+2)+cnt_block)

as a read address at step S16 to read as delay group del2 the bytes stored in a region indicated by this read address and go to processing of step S12.

[0151]

Furthermore, if count value cnt_del by the first counter is set to "3" at step S13, the byte interleaver 12 and the byte de-interleaver 25 each generate the following:

(addr_base+Cx(1+2+3)+cnt_block)

as a read address at step S17 to read as delay group del3 the bytes stored in a region indicated by this read address and go to processing of step S12.

[0152]

If count value cnt_del by the first counter is set to "D-1" at step S13, the byte interleaver 12 and the byte de-interleaver 25 each generate the following:

(addr_base+Cx(1+2+…(D-1))+cnt_block)

as a read address at step S18 to read as delay group del (D-1) the bytes stored in a region indicated by this read address and go to processing of step S12.

[0153]

The byte interleaver 12 performs such an operation by incrementing count value cnt_del from “0” to “D-1” until count value cnt_del becomes “D-1” and repeats this operation as many times as a value of the period C of folding interleaving, to end the processing on all of the bytes included in each packet. The byte de-interleaver 25, on the other hand, performs such an operation by decrementing count value cnt_del from “D-1” to “0” until count value cnt_del becomes “0” and repeats this operation as many times as the value of the period C of the folding de-interleaving, to end the processing on all of the bytes included in each packet.

[0154]

In such a manner, the byte interleaver 12 and the byte de-interleaver 25 each end the processing on the input packet and, if it has been judged at step S12 that there is no input packet, go to processing of step S19. Then, the byte interleaver 12 and the byte de-interleaver 25 respectively increase only once the reference address addr_base by a value of period C of folding interleaving or folding de-interleaving, that is, by one block, at step S19 in the wake of stopping incoming packet and go to processing of step S12 again to wait for the next incoming packet.

[0155]

The byte interleaver 12 and the byte de-interleaver 25 each generate a read address for each delay group in accordance with such rules, and therefore can control data reading from the DPRAM.

[0156]

The byte interleaver 12 and the byte de-interleaver 25, respectively, can perform folding interleaving or folding de-interleaving with a depth of D and a period of C on a packet constituted of N bytes by performing data write and read with respect to the DPRAM in accordance with the rules shown in FIGS. 14 and 15. In this case, typically the byte interleaver 12 and the byte de-interleaver 25 each require such a DPRAM as to have a capacity of DxN bytes, but only need to use such a DPRAM as to have a capacity of Cx(2+3+, ...,+D)= Cx((1+D)xD/2-1). Specifically, in a case of performing folding interleaving or folding

de-interleaving with a depth of 4 and a period of 3 on a packet constituted of 12 bytes as in the example shown in FIGS. 8 to 13, typically the byte interleaver 12 and the byte de-interleaver 25 each require such a DPRAM as to have a capacity of DxN=4x12=48 bytes as described above, but only needs to use such a DPRAM as to have a capacity of Cx(2+3+, ...,+D)= Cx((1+D)xD/2-1)= 3x((1+4)x4/2-1)=27 bytes; further, in a case of performing folding interleaving or folding de-interleaving with a depth of 18 and a period of 12 on a packet constituted of 216 bytes as in the example shown in FIGS. 2 to 7, typically each require such a DPRAM as to have a capacity of DxN=18x216=3888 bytes, but only need to use such a DPRAM as to have a capacity of Cx(2+3+, ...,+D)= Cx((1+D)xD/2-1)= 12x((1+18)x18/2-1)=2040 bytes.

[0157]

In such a manner, the byte interleaver 12 and the byte de-interleaver 25 only need to use such a memory as to have about half a typically required memory capacity, thereby enabling the required memory capacity to be largely reduced.

[0158]

As described above, in a data transmission/reception system described as the embodiment of the present invention, the encoder 10 performs folding interleaving in units of a byte and interleaving in units of a packet, and therefore can achieve a large depth of the interleaving by using a small capacity memory and send transmit data TD that enables even an error correction code with a small code length to correct a significant burst error containing a packet loss. The decoder 20 in the data transmission/reception system, on the other hand, can completely correct a significant burst error containing a packet loss by performing de-interleaving in units of a packet that corresponds to the encoder 10 and folding de-interleaving in units of a byte on the receive data RD received from the encoder 10.

[0159]

It should be noted that the present invention is not limited to the above-mentioned embodiment. For example, although in the above-mentioned

embodiment, the portions including the byte interleaver 12 and the packet interleaver 13 or the portions including the packet de-interleaver 24 and the byte de-interleaver 25 have each been described as being mounted as a programmable device, the present invention can be applied even to a case where they are mounted as a dedicated integrated circuit.

[0160]

Further, although in the above-mentioned embodiment, it has been described that folding interleaving in units of a byte is performed by the byte interleaver 12 and folding de-interleaving in units of a byte is performed by the byte de-interleaver 25, the present invention may perform, as described above, folding interleaving and folding de-interleaving in units of plural consecutive bytes, that is, in units of plural consecutive data words. This is because, for example, in a case where a transport stream TS obtained by performing compression and encoding based on the MPEG-2 standard is input to the encoder 10 and the transport stream TS is restored by the decoder 20, the header of the transport stream TS is of 4 bytes, and accordingly in a case where folding interleaving and folding de-interleaving in units of consecutive 4 bytes are performed, the bytes constituting the header is not rearranged and debug is made easy.

[0161]

In this case, the byte interleaver 12 and the byte de-interleaver 25 each need to be provided with a third counter in addition to the above-mentioned first counter and second counter and, regarding calculation formulae for write address and read address, it is needless to say that count value counted by the third counter is reflected.

[0162]

Further, although in the above-mentioned embodiment, it has been described that the transport stream TS obtained by performing compression and encoding based on the MPEG-2 standard is input to the encoder 10 and the transport stream TS is restored by the decoder 20, the present invention can be applied to any packet communication as far as under such conditions that a

beginning data word of a packet has a fixed value to be used in synchronous generation at the time of folding interleaving or folding de-interleaving in units of a packet and the packet has a fixed length.

[0163]

Furthermore, although in the above-mentioned embodiment, a Reed-Solomon code has been used as the error correction code, the present invention can be applied to encoding and decoding by use of an arbitrary error correction code.

[0164]

Further, although the above-mentioned embodiment has been described with reference to an example where an error is corrected mainly when a packet loss has occurred, of course the present invention can be applied to the case of packet loss occurrence but also to the case where a significant burst error has occurred. Note here that of course the present invention enables an ordinary burst error and a random error to be corrected by performing processing similar to the conventional processing.

[0165]

Furthermore, although the above-mentioned embodiment has been described with reference to a data transmission/reception system provided with the encoder 10 for encoding packeted data and the decoder 20 for decoding data encoded by this encoder 10, the present invention can be applied not only to such encoder and decoder but also to any use where folding interleaving and folding de-interleaving are performed.

[0166]

Of course, the present invention can thus be modified as appropriate as far as it does not depart from its gist.

[0167]

[EFFECT OF THE INVENTION]

As described above in detail, according to the present invention, there is provided an interleaving device replacing and rearranging an order of input data according to a predetermined address for output, and including: a first

interleaving means for performing folding interleaving on first data constituted of plural input packets, in units of a data word or plural consecutive data words; and a second interleaving means for performing folding interleaving on second data constituted of plural packets generated by the first interleaving means, in units of the packet.

[0168]

Accordingly, in such an interleaving device according to the present invention, by successively performing the folding interleaving in units of a data word or plural consecutive data words performed by the first interleaving means and the folding interleaving in units of a packet performed by the second interleaving means, it is possible to generate data in which a significant burst error containing a packet loss can be corrected even with an error correction code having a small code length.

[0169]

Further, according to the present invention, there is provided an interleaving method replacing and rearranging an order of input data according to a predetermined address for output, and including: a first interleaving step of performing folding interleaving on first data constituted of plural input packets, in units of a data word or plural consecutive data words; and a second interleaving step of performing folding interleaving, in units of a packet, on second data constituted of plural packets generated by the first interleaving step.

[0170]

Accordingly, in such an interleaving method according to the present invention, by successively performing the folding interleaving in units of a data word or plural consecutive data words and the folding interleaving in units of a packet, it is possible to generate data in which a significant burst error containing a packet loss even with an error correction code having a small code length.

[0171]

Further, according to the present invention, there is provided a de-interleaving device replacing and rearranging an order of input data according to a predetermined address for output in such a manner as to restore a rearranged

data array using an interleaving device including a first interleaving means for performing folding interleaving on first data constituted of plural input packets, in units of a data word or plural consecutive data words, and a second interleaving means for performing folding interleaving on second data constituted of plural packets generated by the first interleaving means, in units of the packet. The de-interleaving device includes: a first de-interleaving means for performing folding de-interleaving, in units of a packet, on third data constituted of plural input packets, in such a manner as to restore an order of packets of data subjected to the folding interleaving by the second interleaving means to an order of the packets that constitute the second data; and a second de-interleaving means for performing folding de-interleaving, in units of a data word or plural consecutive data words, on fourth data constituted of plural packets generated by the first de-interleaving means, in such a manner as to restore an order of the packets that constitute the second data subjected to the folding interleaving by the first interleaving means to an order of the packets that constitute the first data.

[0172]

Accordingly, in such a de-interleaving device according to the present invention, by successively performing the folding de-interleaving in units of a packet performed by the first de-interleaving means and the folding de-interleaving in units of a data word or plural consecutive data words performed by the second de-interleaving means with respect to data on which the folding interleaving in units of a data word or plural consecutive data words and the folding interleaving in units of a packet have been performed successively, it is possible to completely correct a significant burst error containing a packet loss even with an error correction code having a small code length.

[0173]

Furthermore, according to the present invention, there is provided a de-interleaving method replacing and rearranging an order of input data according to a predetermined address for output in such a manner as to restore a rearranged data array using an interleaving method including a first interleaving step of performing folding interleaving on first data constituted of plural input packets, in

units of a data word or plural consecutive data words, and a second interleaving step of performing folding interleaving, in units of a packet, on second data constituted of plural packets generated by the first interleaving step. The de-interleaving method includes: a first de-interleaving step of performing folding de-interleaving, in units of a packet, on third data constituted of plural input packets, in such a manner as to restore an order of packets of data subjected to the folding interleaving in the second interleaving step to an order of the packets that constitute the second data; and a second de-interleaving step of performing folding de-interleaving, in units of a data word or plural consecutive data words, on fourth data constituted of plural packets generated in the first de-interleaving step, in such a manner as to restore an order of the packets that constitute the second data subjected to the folding interleaving in the first interleaving step to an order of the packets that constitute the first data.

[0174]

Accordingly, in such a de-interleaving method according to the present invention, by successively performing the folding de-interleaving in units of a packet and the folding de-interleaving in units of a data word or plural consecutive data words with respect to data on which the folding interleaving in units of a data word or plural consecutive data words and the folding interleaving in units of a packet have been performed successively, it is possible to completely correct a significant burst error containing a packet loss even with an error correction code having a small code length.

[BRIEF DESCRIPTION OF DRAWINGS]

[FIG. 1] A block diagram for explaining a configuration of a data transmission/reception system shown as an embodiment of the present invention.

[FIGS. 2] Diagrams each for explaining processing of a Reed-Solomon-encoding portion included in an encoder of the data transmission/reception system, in which (A) is a diagram showing a format of a transport packet that is input to the Reed-Solomon-encoding portion, and (B) is a diagram showing a format of an encoded packet generated by the Reed-Solomon-encoding portion.

[FIG. 3] A diagram for explaining processing of a byte interleaver of the encoder, showing a format of encoded data that is input to the byte interleaver and a format of byte interleave data generated by the byte interleaver.

[FIG. 4] A diagram for explaining processing of a packet interleaver of the encoder, showing a format of byte interleave data that is input to the packet interleaver and a format of packet interleave data generated by the packet interleaver.

[FIGS. 5] Diagrams each for specifically explaining transition of data contents in portions in the encoder, in which (A) shows contents of the encoded packet generated by the Reed-Solomon-encoding portion, (B) shows contents of a byte interleave packet generated by the byte interleaver, (C) shows contents of a packet-unit interleave packet generated by the packet interleaver, and (D) shows contents of packet data generated by a packeting portion of the encoder.

[FIG. 6] A diagram for explaining transition of a packet format in the portions in the encoder.

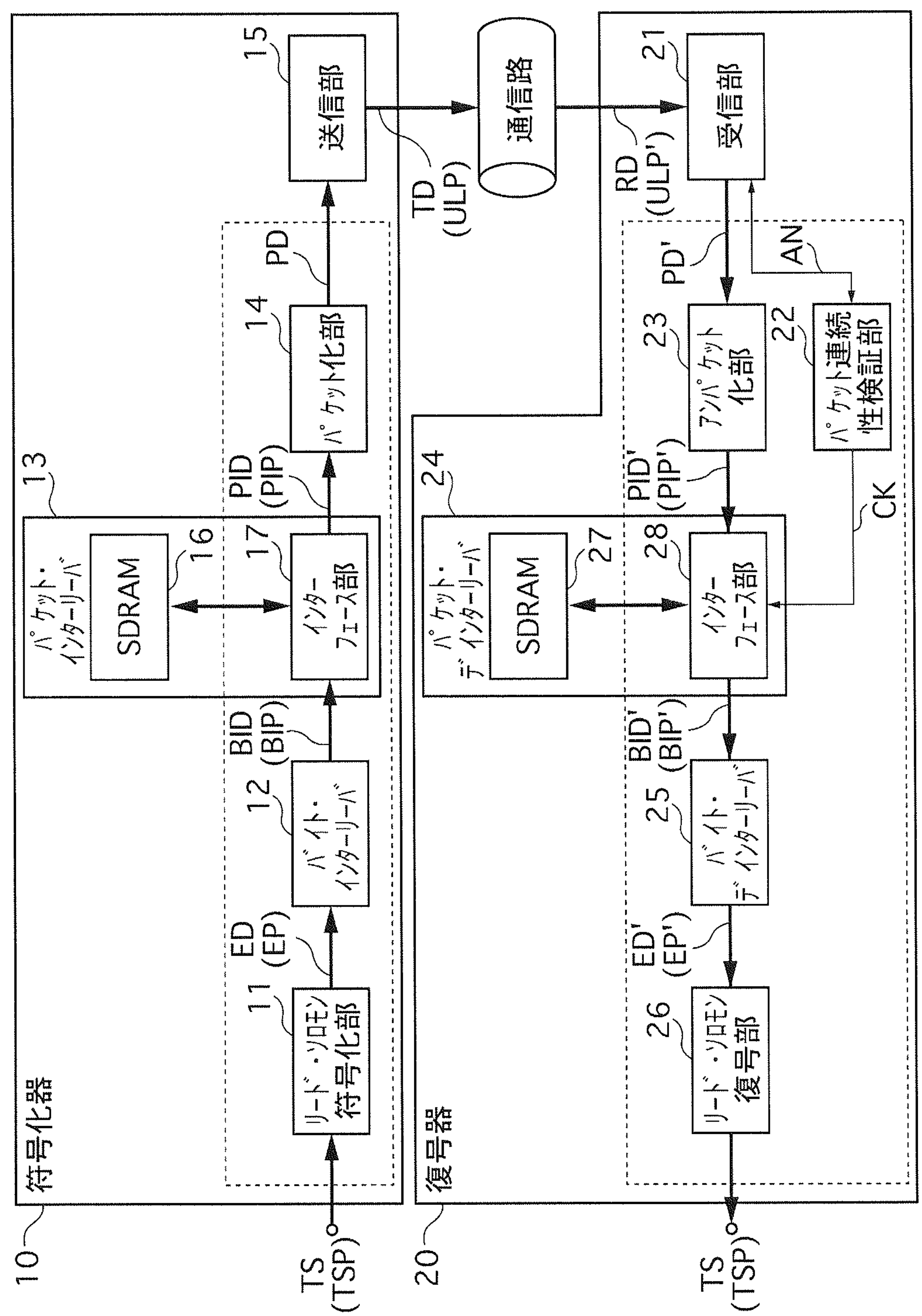
[FIG. 7] A diagram for explaining transition of a packet format in portions of a decoder in the data transmission/reception system.

[FIGS. 8] Diagrams each for explaining the processing of the byte interleaver, in which (A) shows packet Nos. of incoming encoded packets, (B) shows byte Nos. of bytes constituting the encoded packets, (C) shows a count value given by a first counter provided to the byte interleaver, and (D) shows a count value given by a second counter provided to the byte interleaver.

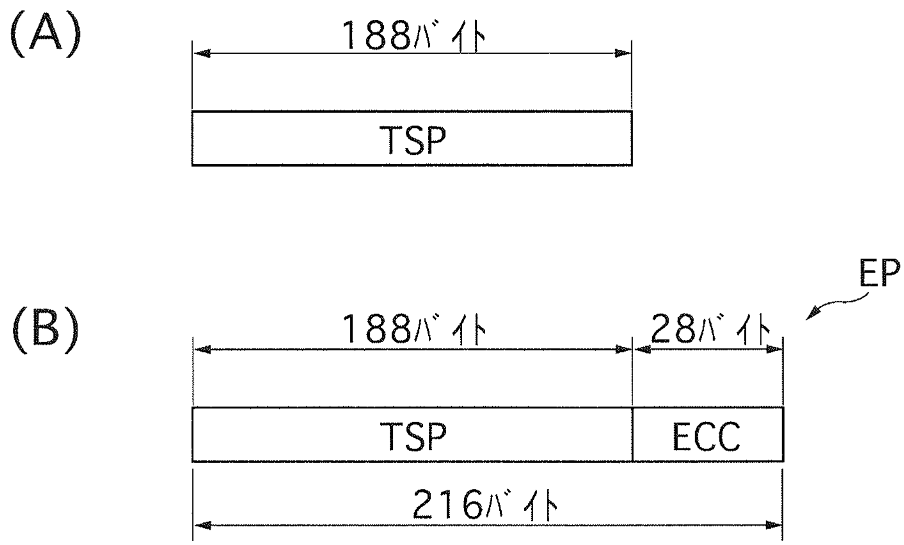
[FIGS. 9] Diagrams each for explaining the processing of the byte interleaver, in which (A) shows packet Nos. of incoming encoded packets, (B) shows a state where the bytes constituting the encoded packets are classified as delay groups, (C) shows a state where data write and read are being performed on a DPRAM provided to the byte interleaver, and (D) shows a state where the bytes of data that is output from the byte interleaver are classified as the delay groups.

[FIG. 10] A diagram showing a state where data is mapped in addresses of the DPRAM provided in the byte interleaver.

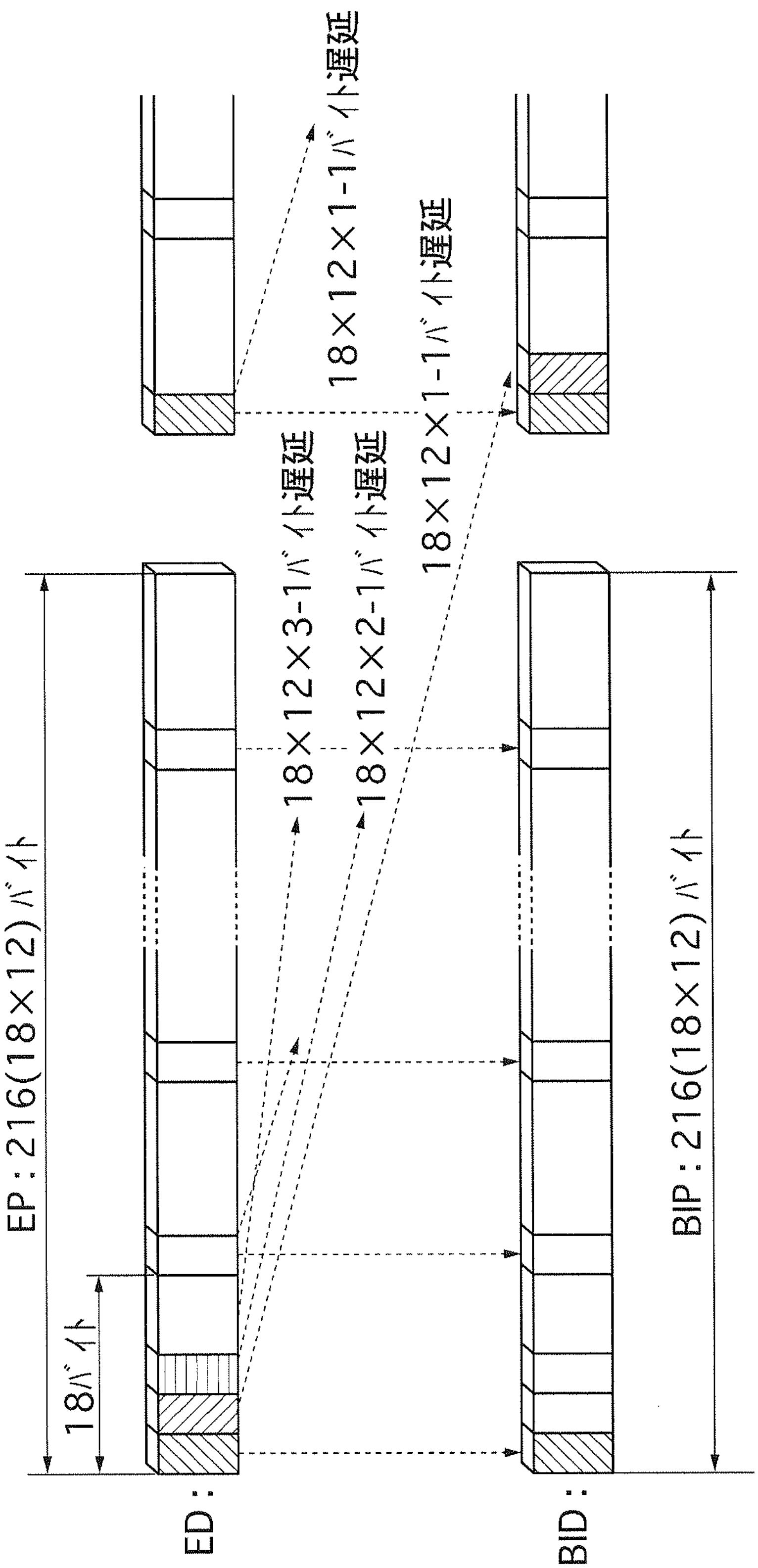
[FIGS. 11] Diagrams each for explaining processing of a byte



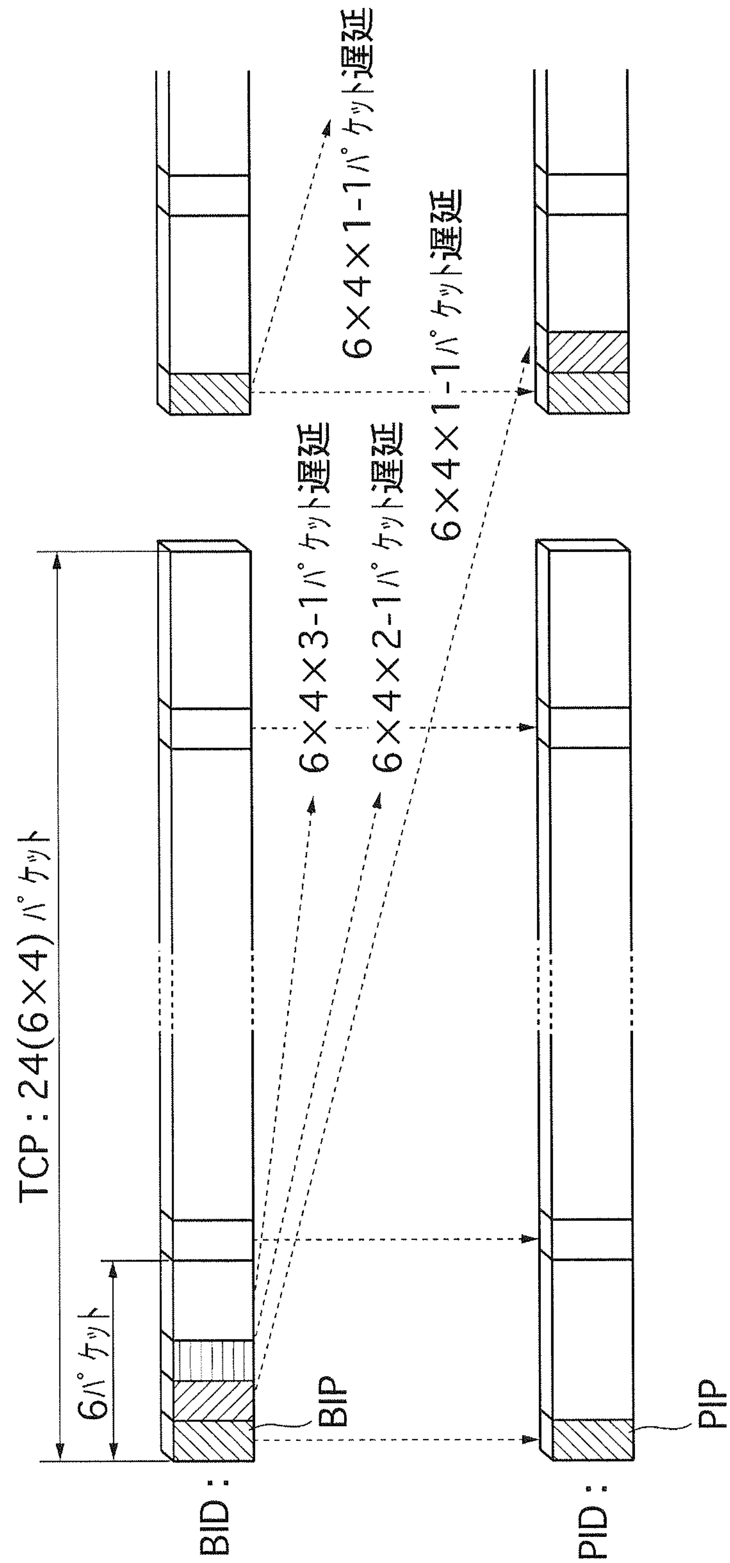
データ送受信システムの構成ブロック図



ﾄﾗﾝｽﾎﾟｰﾄﾊﾟｹｯﾄと符号化ﾊﾟｹｯﾄとの説明図



バイト・インターリーバの処理内容の説明図



パケット・インターリーバの処理内容の説明図

（A）符号化パケットEP

100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136

（B）バイト・インターリーブパケットBIP

100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136
99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132
95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131
94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130
93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129
92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126
89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125
88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124
87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123
86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122
85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121
84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119

18パケット

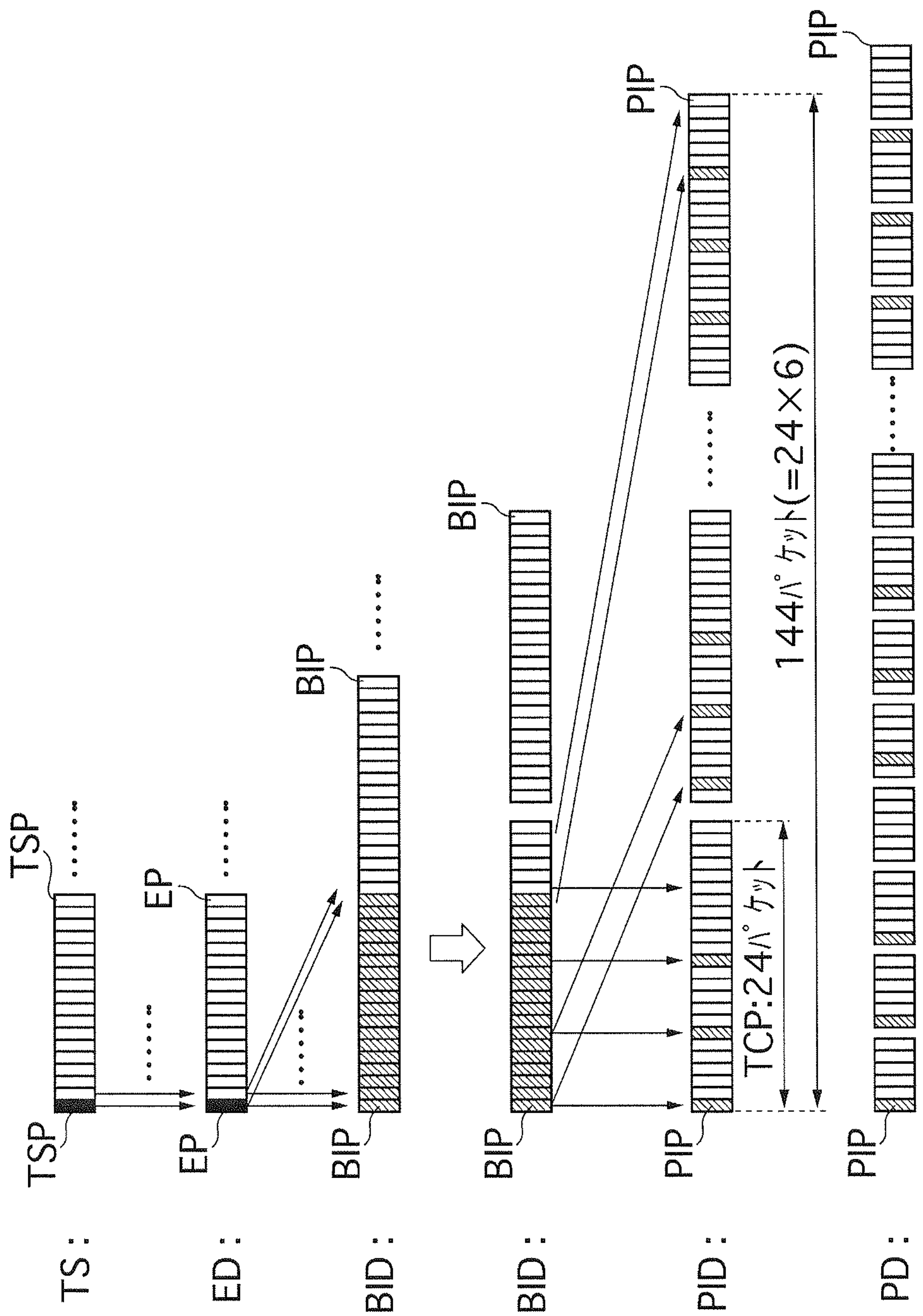
（C）パケット・インターリーブパケットPIP

100	77	54	31	8	-15	106	83	60	37	14	-9	112	89	66	43	20	-3	118	85	72	49	26	3	124	101	78	55	32	9	130	107	84	61	38	15	136
99	76	53	30	7	-16	105	82	59	36	13	-10	111	88	65	42	19	-4	117	94	71	48	25	2	123	100	77	54	31	8	129	106	83	60	37	14	135
98	75	52	29	6	-17	104	81	58	35	12	-11	110	87	64	41	18	-5	116	93	70	47	24	1	122	99	76	53	30	7	128	105	82	59	36	13	134
97	74	51	28	5	-18	103	80	57	34	11	-12	109	86	63	40	17	-6	115	92	69	46	23	0	121	98	75	52	29	6	127	104	81	58	35	12	133
96	73	50	27	4	-19	102	79	56	33	10	-13	108	85	62	39	16	-7	114	91	68	45	22	-1	120	97	74	51	28	5	126	103	80	57	34	11	132
95	72	49	26	3	-20	101	78	55	32	9	-14	107	84	61	38	15	-8	113	90	67	44	21	-2	119	96	73	50	27	4	125	102	79	56	33	10	131
94	71	48	25	2	-21	100	77	54	31	8	-15	106	83	60	37	14	-9	112	89	66	43	20	-3	118	95	72	49	26	3	124	101	78	55	32	9	130
93	70	47	24	1	-22	99	76	53	30	7	-16	105	82	59	36	13	-10	111	88	65	42	19	-4	117	94	71	48	25	2	123	100	77	54	31	8	129
92	69	46	23	0	-23	98	75	52	29	6	-17	104	81	58	35	12	-11	110	87	64	41	18	-5	116	93	70	47	24	1	122	99	76	53	30	7	128
91	68	45	22	-1	-24	97	74	51	28	5	-18	103	80	57	34	11	-12	109	86	63	40	17	-6	115	92	69	46	23	0	121	98	75	52	29	6	127
90	67	44	21	-2	-25	96	73	50	27	4	-19	102	79	56	33	10	-13	108	85	62	39	16	-7	114	91	68	45	22	-1	120	97	74	51	28	5	126
89	66	43	20	-3	-26	95	72	49	26	3	-20	101	78	55	32	9	-14	107	84	61	38	15	-8	113	90	67	44	21	-2	119	96	73	50	27	4	125
88	65	42	19	-4	-27	94	71	48	25	2	-21	100	77	54	31	8	-15	106	83	60	37	14	-9	112	89	66	43	20	-3	118	95	72	49	26	3	124
87	64	41	18	-5	-28	93	70	47	24	1	-22	99	76	53	30	7	-16	105	82	59	36	13	-10	111	88	65	42	19	-4	117	94	71	48	25	2	123
86	63	40	17	-6	-29	92	69	46	23	0	-23	98	75	52	29	6	-17	104	81	58	35	12	-11	110	87	64	41	18	-5	116	93	70	47	24	1	122
85	62	39	16	-7	-30	91	68	45	22	-1	-24	97	74	51	28	5	-18	103	80	57	34	11	-12	109	86	63	40	17	-6	115	92	69	46	23	0	121
84	61	38	15	-8	-31	90	67	44	21	-2	-25	96	73	50	27	4	-19	102	79	56	33	10	-13	108	85	62	39	16	-7	114	91	68	45	22	-1	120
83	60	37	14	-9	-32	89	66	43	20	-3	-26	95	72	49	26	3	-20	101	78	55	32	9	-14	107	84	61	38	15	-8	113	90	67	44	21	-2	119

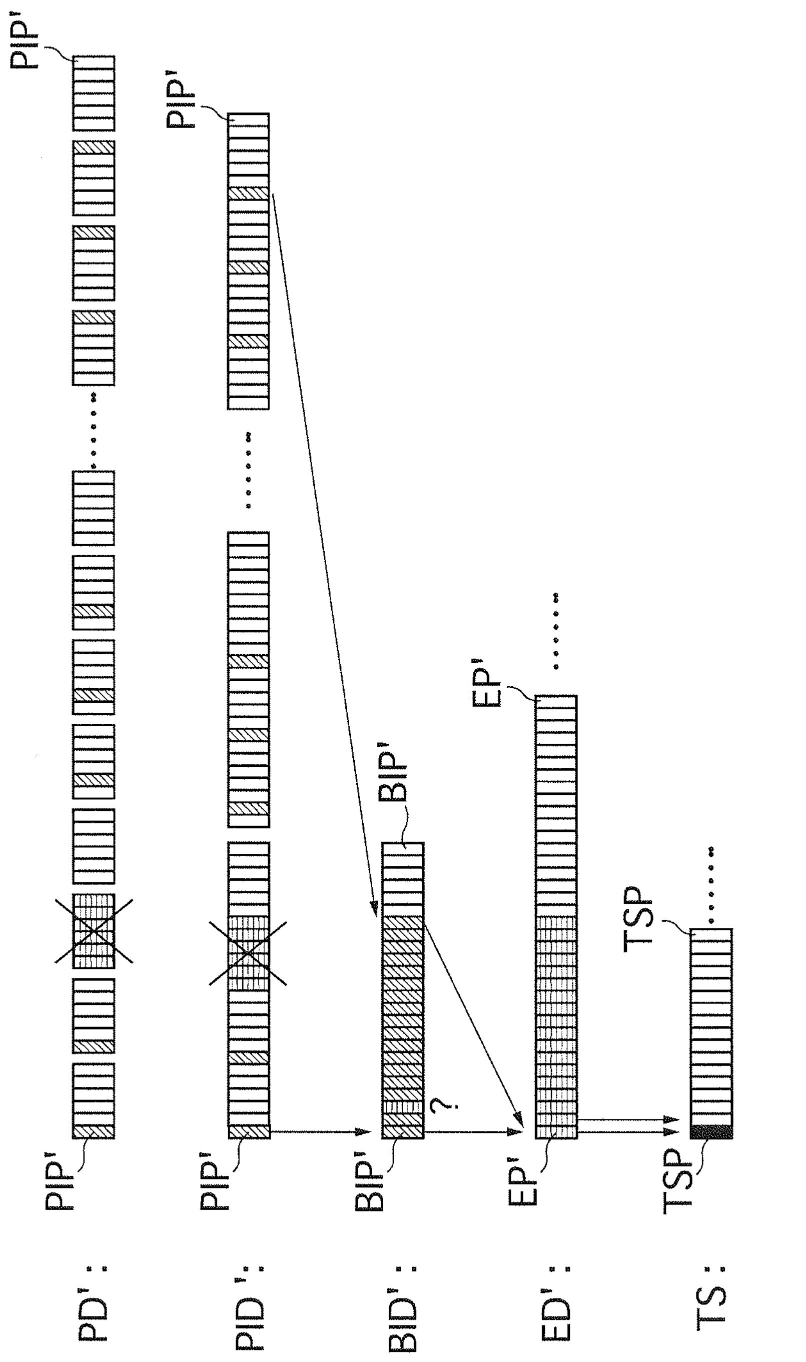
（D）パケットデータPD

118	85	72	49	26	3
117	94	71	48	25	2
116	93	70	47	24	1
115	92	69	46	23	0
114	91	68	45	22	-1
113	90	67	44	21	-2
112	89	66	43	20	-3
111	88	65	42	19	-4
110	87	64	41	18	-5
109	86	63	40	17	-6
108	85	62	39	16	-7
107	84	61	38	15	-8
106	83	60	37	14	-9
105	82	59	36	13	-10
104	81	58	35	12	-11
103	80	57	34	11	-12
102	79	56	33	10	-13
101	78	55	32	9	-14

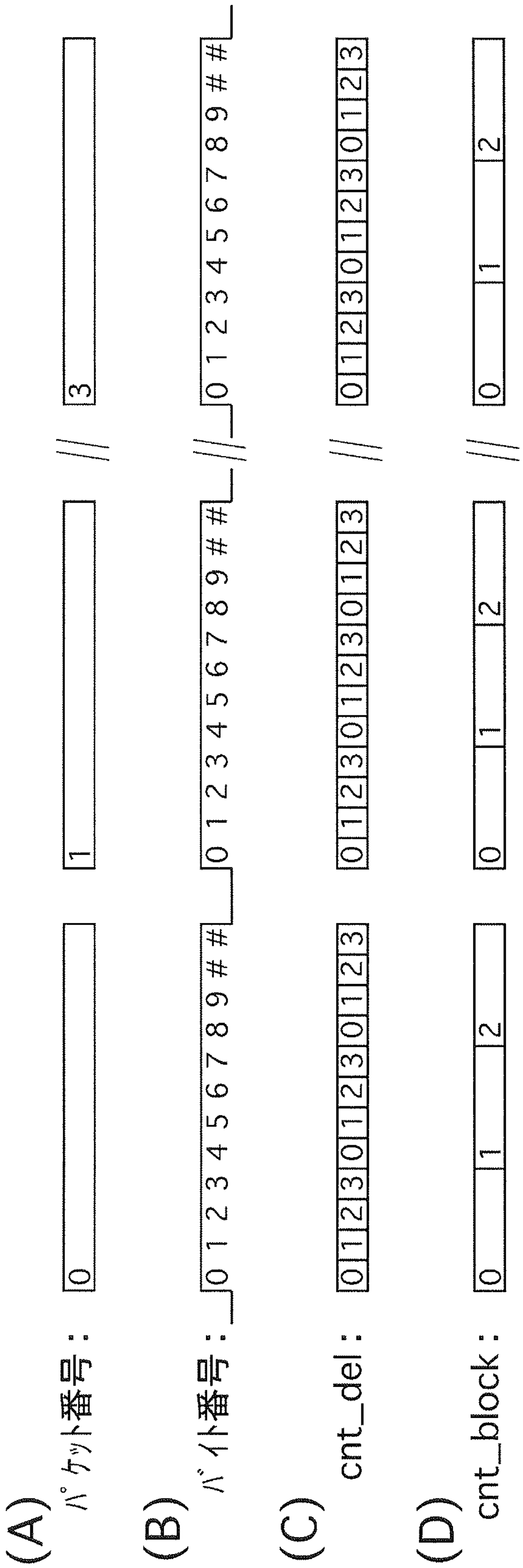
各部におけるデータ内容の推移の説明図



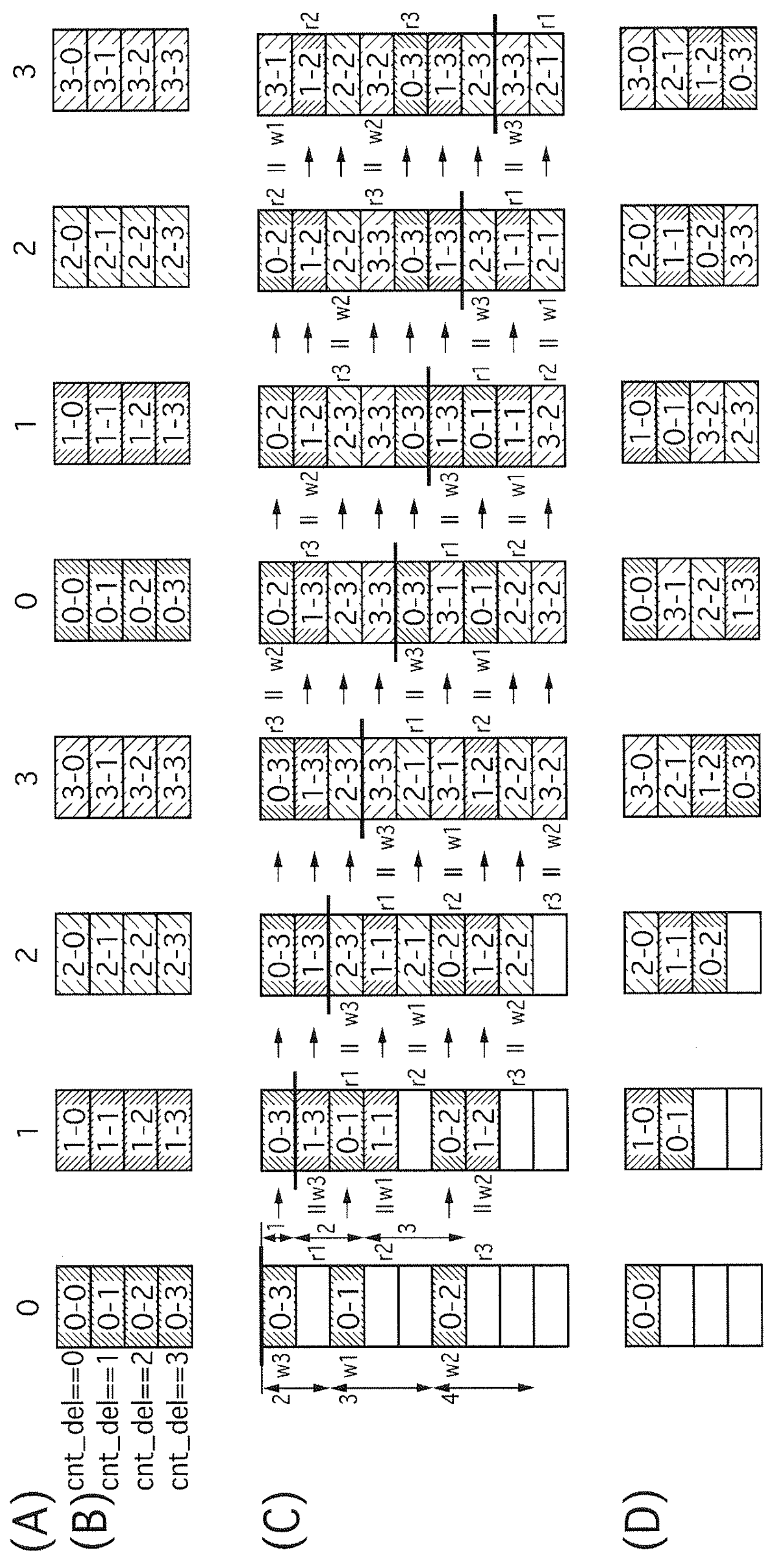
パケットフォーマットの説明図



パケットフォーマットの説明図



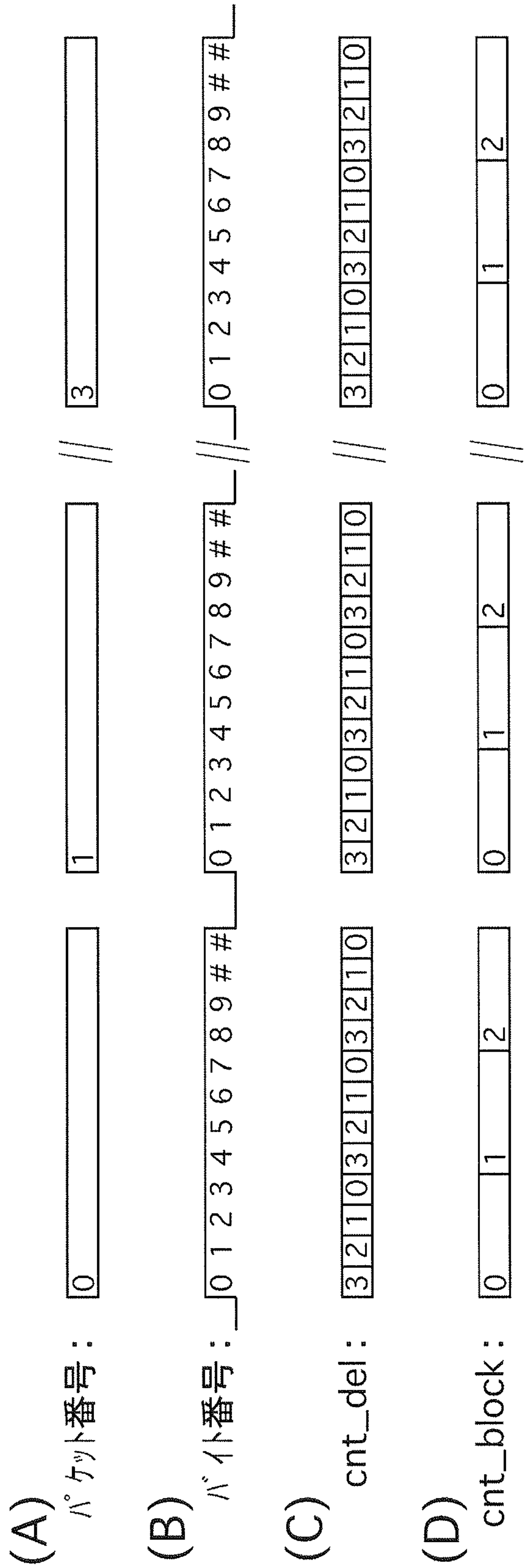
バイト・インターリーバの処理内容の説明図



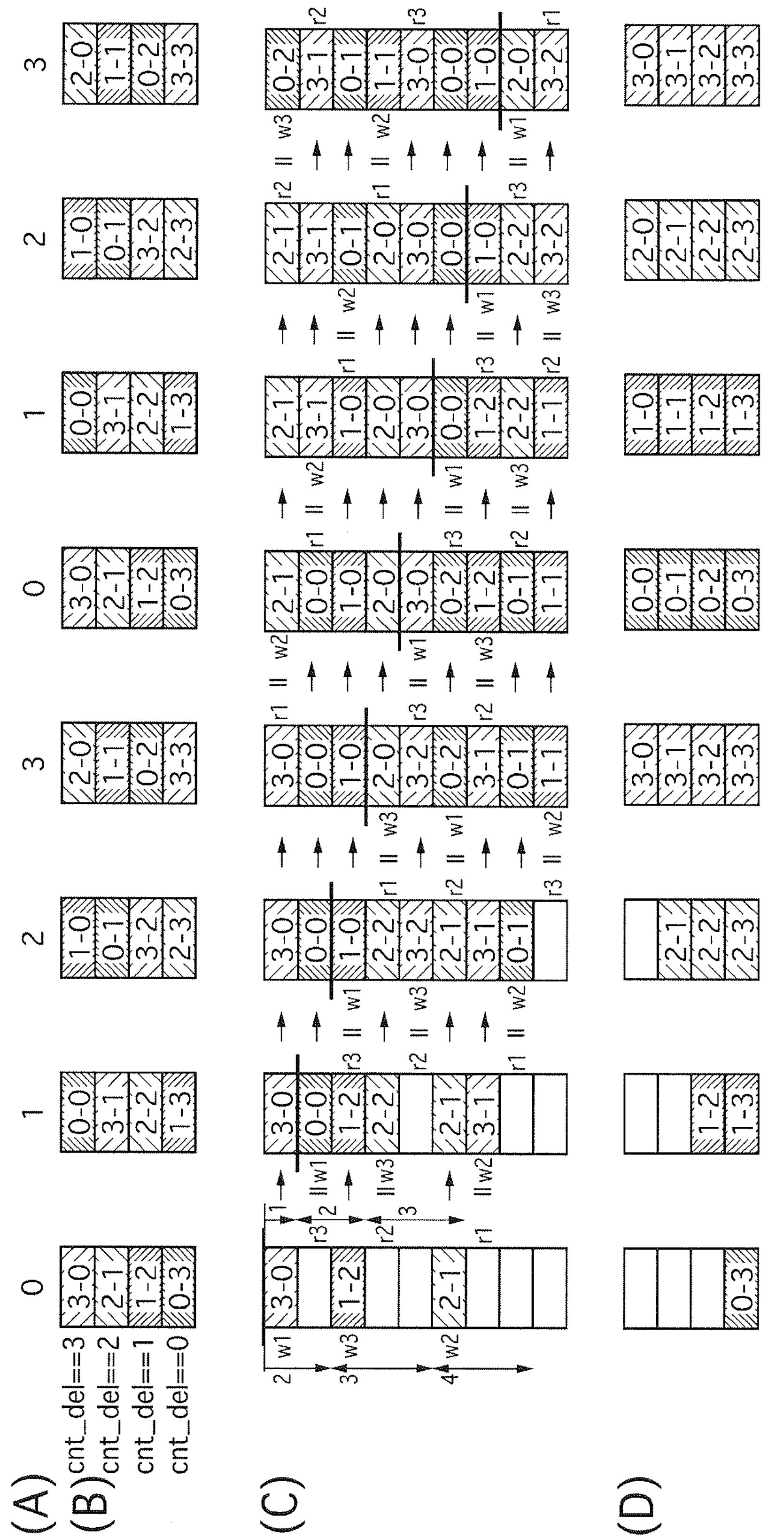
バイト・インターリーバの処理内容の説明図

ｱﾄﾞﾚｽ	ﾊﾟｹｯﾄ番号	遅延ﾊﾟｹｯﾄ数	ﾊﾞｲﾄ番号
0	0	3	3
	0	3	7
	0	3	11
3	1	3	3
	1	3	7
	1	3	11
6	2	3	3
	2	3	7
	2	3	11
9	3	3	3
	3	3	7
	3	3	11
#	2	1	1
	2	1	5
	2	1	9
#	3	1	1
	3	1	5
	3	1	9
#	1	2	2
	1	2	6
	1	2	10
#	2	2	2
	2	2	6
	2	2	10
#	3	2	2
	3	2	6
	3	2	10
#			

ﾒﾓﾘﾏｯﾌﾟの説明図



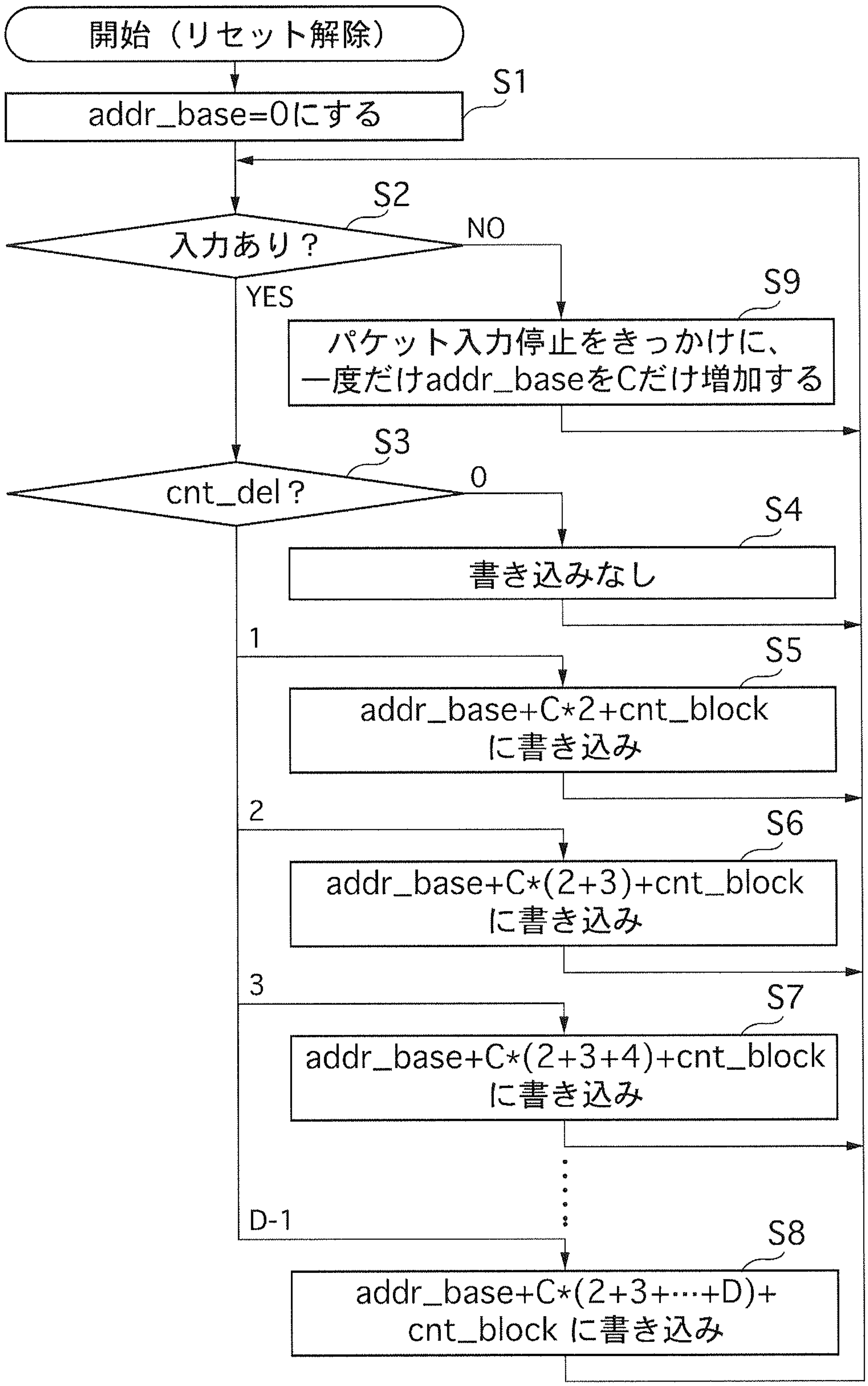
バイト・デインターリーバの処理内容の説明図



バイト・デインターリーバの処理内容の説明図

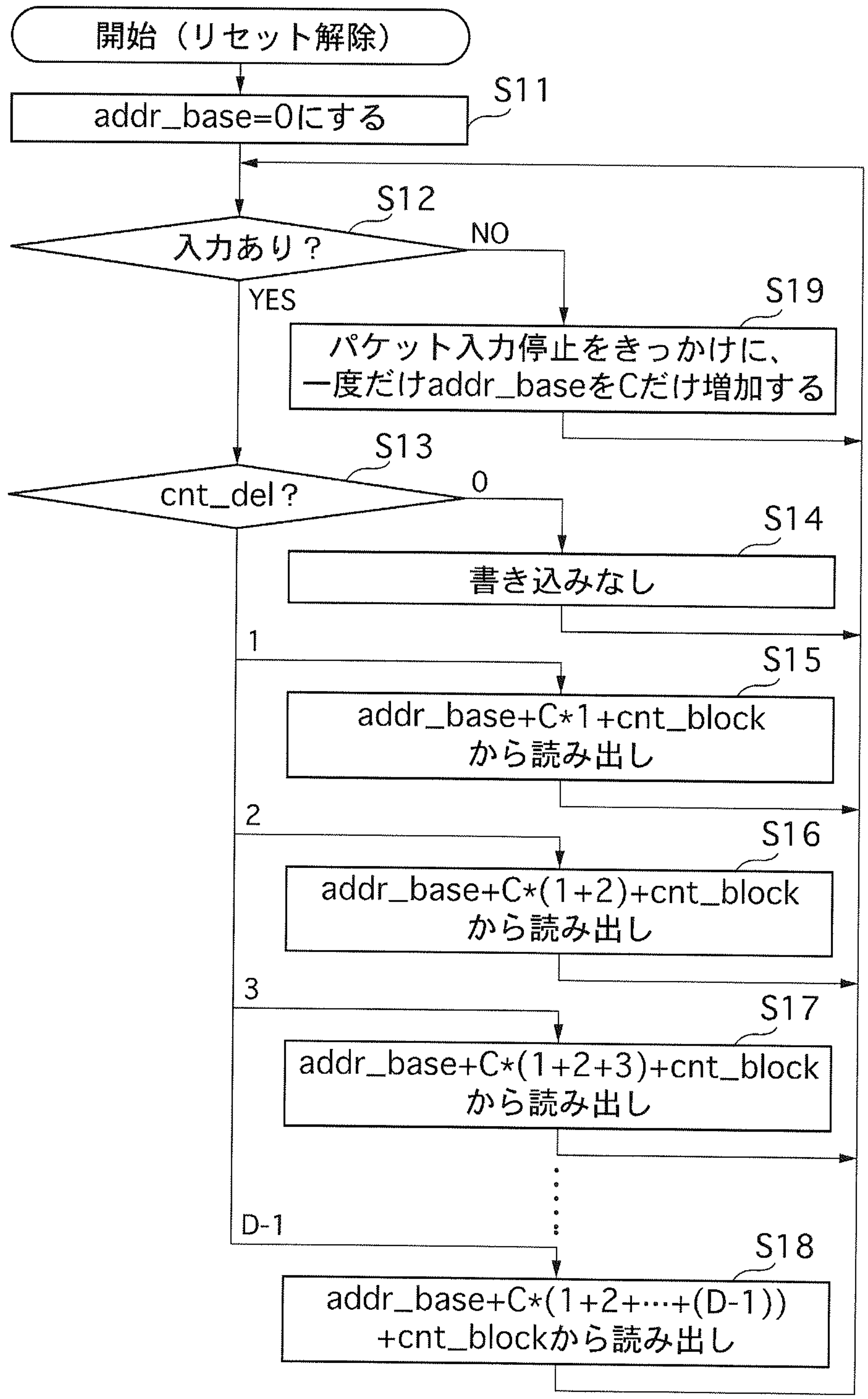
ｱﾄﾞﾚｽ	ﾊﾟｹｯﾄ番号	遅延ﾊﾟｹｯﾄ数	ﾊﾞｲﾄ番号
0	3	3	0
	3	3	4
	3	3	8
3	0	3	0
	0	3	4
	0	3	8
6	1	3	0
	1	3	4
	1	3	8
9	2	3	0
	2	3	4
	2	3	8
#	3	1	2
	3	1	6
	3	1	10
#	0	1	2
	0	1	6
	0	1	10
#	3	2	1
	3	2	5
	3	2	9
#	0	2	1
	0	2	5
	0	2	9
#	1	2	1
	1	2	5
	1	2	9
#			

ﾒﾓﾘﾏｯﾌﾟの説明図



書き込みｱﾄﾞﾚｽ生成の際の一連の処理工程

【図15】



読み出しｱﾄﾞﾚｽ生成の際の一連の処理工程

【書類名】図面 [NAME OF DOCUMENT] DRAWINGS

【図１】 FIGURE 1

10 Encoder

11 Reed-Solomon-encoding portion

12 Byte interleaver

13 Packet interleaver

14 Packeting portion

15 Transmission portion

17 Interface portion

21 Reception portion

22 Packet consecutiveness verification portion

23 Un-packeting portion

24 Packet de-interleaver

25 Byte de-interleaver

26 Reed-Solomon-decoding portion

28 Interface portion

通信路 Communication path

データ送受信システムの構成ブロック図 Block diagram of configuration of data transmission/reception system

【図２】 FIGURES 2

バイト Byte

トランスポートパケットと符号化パケットとの説明図

Explanatory diagrams of transport packet and encoded packet

【図３】 FIGURE 3

バイト Byte

バイト遅延 Byte delay

バイト・インタリーバの処理内容の説明図

Explanatory diagram of processing contents of byte interleaver

【図４】 FIGURE 4

パケット Packet

パケット遅延 Packet delay

パケット・インタリーバの処理内容の説明図

Explanatory diagram of processing contents of packet interleaver

【図５】 FIGURES 5

(A) Encoded packet

(B) Byte interleave packet

(C) Packet-unit interleave packet

(D) Packet data

パケット Packet

各部におけるデータ内容の推移の説明図

Explanatory diagrams of transition of data contents in portions

【図６】 FIGURE 6

パケット Packet

パケットフォーマットの説明図 Explanatory diagram of packet format

【図７】 FIGURE 7

パケットフォーマットの説明図 Explanatory diagram of packet format

【図８】 FIGURES 8

パケット番号 Packet No.

バイト番号 Byte No.

バイト・インターリーバの処理内容の説明図

Explanatory diagrams of processing contents of byte interleaver

【図９】 FIGURES 9

バイト・インターリーバの処理内容の説明図

Explanatory diagrams of processing contents of byte interleaver

【図１０】 FIGURE 10
アドレス Address
パケット番号 Packet No.
遅延 Delay
バイト番号 Byte No.
パケット数 Number of packets
メモリマップの説明図 Explanatory diagram of memory map

【図１１】 FIGURES 11
パケット番号 Packet No.
バイト番号 Byte No.
バイト・デインターリーバの処理内容の説明図
Explanatory diagrams of processing contents of byte de-interleaver

【図１２】 FIGURES 12
バイト・デインターリーバの処理内容の説明図
Explanatory diagrams of processing contents of byte de-interleaver

【図１３】 FIGURE 13
アドレス Address
パケット番号 Packet No.
遅延 Delay
バイト番号 Byte No.
パケット数 Number of packets
メモリマップの説明図 Explanatory diagram of memory map

【図１４】 FIGURE 14
S1 Set addr_base to 0
S2 Is there an input?
S4 No write

S5 to S8 Write to addr_base+C……

S9 Increase only once addr_base by C in the wake of stopping incoming packet

開始（リセット解除） Start (cancel of reset)

書き込みアドレス生成の際の一連の処理工程

A series of processing steps in generating write address

【図１５】 FIGURE

S11 Set addr_base to 0

S12 Is there an input?

S14 No write

S15 to S18 Read from addr_base+C……

S19 Increase only once addr_base by C in the wake of stopping incoming packet

開始（リセット解除） Start (cancel of reset)

読み出しアドレス生成の際の一連の処理工程

A series of processing steps in generating read address

[NAME OF DOCUMENT] ABSTRACT

[SUMMARY]

[OBJECT]

To enable correction of a significant burst error containing a packet loss even with an error correction code having a small code length.

[MEANS FOR SOLVING]

An encoder 10 in a data transmission/reception system has a byte interleaver 12 for performing folding interleaving on encoded data ED in units of a byte and a packet interleaver 13 for performing folding interleaving, in units of a packet, on byte interleave data BID generated by this byte interleaver 12. On the other hand, a decoder 20 has a packet de-interleaver 24 for performing folding de-interleaving, in units of a packet, on packet interleave data PID' and a byte de-interleaver 25 for performing folding de-interleaving, in units of a byte, on byte interleave data BID' generated by this packet de-interleaver 24.

[SELECTED DRAWING] Figure 1